AD-A230 050

S P M

Version 1.1

1990

The Charles Stark Draper Laboratory, Inc.

555 Technology Square, Cambridge, Massachusetts 02139
Mail Station 1E

Telephone (617) 258-2914

S P M

Version 1.1

The Charles Stark Draper Laboratory, Inc.

555 Technology Square, Cambridge, Massachusetts 02139 Telephone (617) 258-2914
Mail Station 1E

S P M

Version 1.1

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

The Charles Stark Draper Laboratory, Inc.

555 Technology Square, Cambridge, Massachusetts 02139
Mail Station 1E

Telephone (617) 258-2914

Abstract

Richard E. Gustavson
Technical Staff Member
C. S. DRAPER LABORATORY, INC.
Cambridge, MA 02139

S. P. M. : A Connection from Product to Assembly System

A major thrust of our (consulting and research) work in concurrent design of products and processes has been to create a direct link between product specifications and the required methods for assembly. Using the totally-assembled geometry and component mate data as the basis, we can now establish an exploded view, the assembly ***S***equence, the assembly process ***P***lan, and the assembly system task / resource ***M***atrix (which is required input for assembly system design software). This process has been captured in the S.P.M. software, currently implemented on the IBM PC or compatibles. All phases of the program allow user modifications (e.g. alter view order, change sequence, add/delete process steps, alter time/safety factors, specify programmable resource types). The process is usable for the range of initial concept design to actual production requirements. This work was supported in part by DARPA Contract MDA972-88-C-0027.

S. P. M. : A Connection from Product to Assembly System

INTRODUCTION

While a product is being designed, it is desirable to obtain a reasonable idea of how it can be fabricated and assembled in order to minimize the life-cycle costs. One of the goals of our work has been to provide "expert" manufacturing engineering to the process of determining an assembly system design. PRODUCT design engineers should also be able to benefit. Considerable work on fabrication systems has been done by others and will not be addressed here. We are primarily interested in creating usable, economically viable assembly systems (which are the integrator of all other manufacturing processes). Figure 1 exhibits highlights of the process required. Generally, we need to know how to take certain product information as a basis, determine an assembly sequence, produce an appropriate process plan, and convert that into tasks and applicable resources. This paper describes a method (SPM) for performing these steps and an interactive computer program for carrying them out. The resulting data forms most of the input required for synthesizing assembly systems using a method such as A.S.D.P. (1). The combination of SPM and ASDP provides designers with a convenient and fast tool for converting product design data into concept assembly system designs and assembly cost data.

A fundamental assumption is that the breakdown of a product into subassemblies occurs **only** on a functional basis and usually results in non-separable, easily-tested units. Starting with the final assembly, we can define its components (which may be individual parts or sub-assemblies). Each of those sub-assemblies can be broken down in a similar manner. The process repeats until all individual parts of the total product have been enumerated. What follows assumes that such choices have been logically made and that we are dealing (at any one time) uniquely with the components of only ONE such unit. We have strong industrial evidence that the largest practical system is required to assemble no more than **twenty-four** components; it is desirable that every product be broken down into "sub-assemblies" that satisfy this size criterion.

We assume that any applicable DfA/DfM methods have already been applied (2-6).

Many researchers are investigating assembly sequences (7-19). A fundamental premise in the analyses is that all mathematically possible combinations should be enumerated; this significantly large number is reduced by various means (usually avoidance of difficult or undesirable conditions). Most groups outside Draper Laboratory pursue this information with the intention of finding alternatives to a prescribed sequence such that a part shortage or sub-assembly failure (for example) can be overcome at a flexible assembly station; this procedure can be called "opportunistic scheduling". Our group looks for complete assembly sequences only; we seek to design total assembly systems for manufacturers.

The method described here provides a significant assist to a product design team in establishing a usable sequence (which is the first step in creating an assembly system). It is the author's contention that **all** useful assembly sequences for a given sub-assembly are minor variations of each other and that they number about **four**. (This is not a function of the number of components but of the number of multi-axis mates.) Be advised that significant differences in fixturing requirements and/or re-orientations are possible.

The mathematical relationships shown are all curve-fit from actual data that we have derived and used in our consulting work. Most of the product-related equations are surprisingly linear. Task and resource data requires more complex relationships.

SPM provides a product designer with important insight into the complex world of assembly systems **and** meaningful information about the effect of product design on them. Concurrently, a manufacturing engineer has access to significant system design knowledge which is very difficult to obtain by any other means. It is worth noting that the SPM user has numerous opportunities for altering the "expert" results which are established at various stages of the procedure (shown schematically in Figure 2).

Let's have a closer look at Figure 2. There are four major sections to SPM. Starting at the lower left and working clockwise, we have:

> The starting point is bounding-box geometry and mate data for the assembly.

Next, we establish a usable assembly sequence as a result of creating an exploded view of the assembly. (section S)

Then, we specify intermediate test requirements (where applicable) and produce the assembly process plan. (section **P**)

Finally, we determine which generic resource types (manual, fixed or programmable) are applicable to each task, establish a specific group of programmable resource types for this assembly, and fill in the task/resource-type matrix. (section M)

PRODUCT DATA REQUIREMENTS

The first step in the procedure is to specify the product information that will be required. It may be input manually or extracted from a CAD data base. As an example, Figure 3 shows an Air Conditioning Module (ACM) while Figure 4 exhibits the required component data which is obtained from the **totally-assembled location** of all components. A convenient orthogonal coordinate system (with the major axis of assembly always being designated as Z) must be specified.

For each component, we need to know :

weight (used to determine size and cost of resources),

a single character (usually a letter) code designator,

a description (helps casually-involved users),

and six bounding plane dimensions specified as follows (helps create an approximate component picture, quickly) :

Minimum X	Maximum X
Minimum Y	Maximum Y
Minimum Z	Maximum Z

These component data records can be entered in any convenient order; they need not be in any geometric or alphabetic pattern.

Using this data, we can readily determine which component has the highest Z plane (denoted the TOP) and which has the lowest Z plane (denoted the BOTTOM). All other components are called MIDDLE.
We also need information about the **physical** mates which occur. Figure 5 exhibits the data for the Air Conditioning Module.

Characteristics of representative products are:

Example	Components	Mates	Figure
Generic Gimbal	11	16	*
Large Trans. Final Assy.	12	20	13
Seeker Head Assembly	17	24	12
Seeker Head - A	7	8	#
Seeker Head - B	11	16	#
Complete A. C. M.	15	26	3
Optics Sub-assembly	16	21	10
Simplified TRAX Final Assy.	12	18	15
Trans-axle Final Assy.	22	37	14
Disk Head Assembly	16	18	9
Solid Rear Axle Assembly	17	18	11

* Simpler version of Seeker Head - B

\# Portion of total Seeker Head Assembly (Fig.12)

When this data is graphed as Figure 6, we see that we can expect

$$m = 1.1\ c^{1.1}$$

where

m = number of mates

c = number of components

The required mate data includes :

the code identifying each of the two mating components (note that any component can mate with many others) ,

the mate type (see list below),

the number of physical actions (e.g. multiple gears, screws in a pattern),

and the fit requirements (where applicable).

This information is converted within the SPM computer program into the format shown in Figure 7, which contains the following:

A. Mating component codes.

B. Type of mate selected from the following ten mate types:

	Category	name	DoD	MD
1	Adhesively bonded	N	3	1
2	Bearing race / bushing	A	df	1
3	Bolted joint(s) (without specification)	T	1	1
4	Critical alignment	L	3	1
5	Gear(s) mate	A	df	2
6	General placement	P	1	2
7	Selective fit	G	2	1
8	Snap fit	E	1	1
9	Spline(s) mate	A	df	2
10	Torqued fastener(s) (to specification)	B	2	1

where **DoD** means Degree of Difficulty of the mate (see C below).

df means that the **DoD** depends upon the fit specified and the number of mates (note in C below that this value can turn out to be as high as 4).

MD signifies the number of fine motion directions required for assembly (see D below).

C. Degree of difficulty of the mate - a measure of the complexity of the task. It attempts to account for such conditions as tightness of fit, limited accessibility, multiplicity of mates, two-handed operation, etc.

1 : Straightforward task.

2 : Some trouble possible.

3 : Hard to do; fixed automation not possible.

4 : Arduous; must be done manually.

Although pre-defined (using proprietary methods) for each mate type, the degree of difficulty can be modified (in the process planning section) if desired.

D. Number of fine-motion assembly directions required - the degrees-of-freedom (translation and/or rotation) required to perform the final action. While it is desirable to reduce this number to one, it is not always possible (e.g. gears, splines, general motion).

E. Number of (simultaneous) events which must take place - this value is one for almost everything except such conditions as planetary gears or threaded fasteners in a pattern.

F. Number of (probably inseparable) sub-tasks which have been collected and specified as one - this usually occurs when a group of components must be assembled before the collection is added to the main assembly. The value is generally determined by adding 1 to the number of semi-colons in the parts-to-be-assembled description (which must currently be input by the user - see Figure 4). Under some circumstances, this group could be a separate sub-assembly and would therefore be subject to the SPM process).

An important parameter, later used to establish the relative assembly difficulty, is the sum of the degrees-of-difficulty (DoD) for all components of the product / sub-assembly. Using the sums found for the assemblies in the table above, we find that this DoD sum (s) can be reasonably approximated by

$$s = 2.3\,m - 4.3$$

We can get a good *first approximation* about the product's difficulty of assembly from the following:

Equation	Mode
$s < (3.2\,c - 14) = S(E)$	Easy
$(3.2c - 14) <= s <= (3.2c + 10)$	Moderate
$s > (3.2\,c + 10) = S(D)$	Difficult

For example, suppose that a sub-assembly has twenty components (c=20). If the sum of the degrees-of-difficulty (s) is less than 50, the assembly is expected to be rather simple. If s exceeds 74, we can anticipate trouble for some of the tasks. It might seem that further sub-division of such an assembly would reduce complexity; the following table shows that sub-dividing the 20 components into groups of 8 and 12 does not necessarily produce that result:

c	m	s	S(E)	S(D)	Mode
20	30	64	50	74	Moderate
12	17	35	24	48	Moderate
8	11	21	12	36	Moderate

ASSEMBLY SEQUENCE PROCEDURE (Step S)

Establishing a usable assembly sequence has turned out to be a rather simple procedure. After determining the base component, we can develop an exploded view of all other components relative to it. That picture will provide (probably after some minor modifications) the basis for the general sequence. Depending upon the complexity of the product / sub-assembly and/or the need for compound mates, a number of modifications will be made by the user. In a few minutes (assuming the data files already exist), a usable sequence can be established using the following procedure.

Determine the base component

We use a simple method for determining the base (initially placed) component. Each component in the assembly (Figure 4) is scored according to its weight (W), its enclosed rectangular volume (V), and the sum (S) of the degree-of-difficulty of its mates (the larger the value, the smaller the score). Note that the absolute value is used to determine score and that the lower the score, the better the characteristic (i.e. 2 is better than 6). The sum of these three scores is determined for each component. The one with the **lowest** total score is the base component.

This general procedure has almost always produced a usable base component. At least two special cases arise:

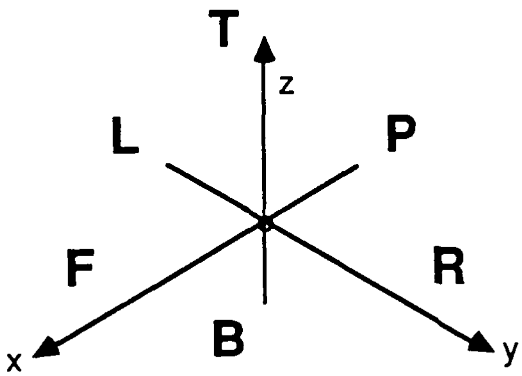
1. If a component scores highest for S and for at least one of the other characteristics (W or V), then it will be designated the base (regardless of its total score).

2. When the sets of physical mates for the TOP and BOTTOM components contain a common component, that will be the base. This is commonly the base component , anyway.

We shall see later that a simple sequence alteration scheme allows rearrangement of all tasks; at this point, we seek only a good initial (base) component which allows us to draw a picture of the product.

Establish an Exploded View of the Assembly

There are assumed to be six possible main directions for assembly (**T**op, **F**ront, **L**eft, **P**osterior, **R**ight and **B**ottom) as shown below:



Note that each component has an assembly direction (with respect to the base component) which is determined by merely checking all of its mates and establishing which mate direction (X, Y, or Z) contains the largest number. For each component, we determine whether its location is positive or negative by computing where the component's **geometric** center (defined as being halfway between the bounding planes for all axes) is located relative to the base component's geometric center (in the fully-assembled condition). The position along that axis will be identified (for drawing purposes) by the plane on that axis which is furthest away (e.g. a -Z component uses its minimum Z plane dimension). This allows us to easily order the components in each of the six possible directions. Figure 8 shows the initial ACM exploded view found using this method; it does an amazingly good job of determining stack-up order. How well it works is a function of the geometric complexity of the product as the following table shows:

Product	Mis-ordered Components	Total	Figure
Air Conditioning Module Assy.	2	15	3
Disk Head Assembly	3	16	9
Optics Sub-assembly	3	16	10
Solid Rear Axle Assembly	2	17	11
Seeker Head Assembly	2	17	12
Large Transmission Assy.	2	12	13
Trans-axle Assy. (total)	6	22	14
Trans-axle Assy. (major comp.)	2	12	15

None of these products is so "simple" that we could expect our initial ordering to be perfect. Most of these mis-placements are due to :

> the geometric center being on the opposite side of the actual assembly direction ,

> or, the component having an unusual shape wherein the "away" bounding plane causes it to appear in a peculiar place.

We have found that the approximate number of exploded view changes (e) that will be required is

$$e = \frac{c - 7}{4}$$

Even though initial discrepancies often exist, the software makes it very easy to change the picture such that reality is better represented. For the ACM example, we have to move two components ([S] and [N], see Figures 8 and 16) since they must be assembled through the "open" top of the case [L] even though they end up near the "solid" bottom of it. Components on each axis can be put <u>between</u> any two <u>consecutive</u> other components on that general axis (positive and negative are not distinguished, but observe that the base component appears on the X-axis and Y-axis as well as on the Z-axis).

Once the exploded view is satisfactory, we are ready to establish the assembly sequence.

Define the Assembly Sequence

Figure 17 exhibits another exploded view of the ACM product as well as a list of the components in each of the six (where applicable) assembly directions. Since there is currently no knowled base (relating to this subject) available, the user becomes the expert" and must specify the general direction order. For this example, the "expert" has selected the order Bottom, Left, Posterior and Top (there are *no Right or Front components).*

The result of that direction choice is shown in the following table:

```
          Current Assembly Sequence

 1 (L) Evaporator C 15 (A) Cover
 2 (P) AI Case; AI
 3 (B) MVH Sub-Assy
 4 (F) Temp. Valve
 5 (O) Solenoid  #1
 6 (E) Vacuum Eleme
 7 (J) Pipe Seal
 8 (U) Harness
 9 (K) Resistor Ass
10 (T) Solenoid Ass
11 (S) Temperature
12 (H) Motor & Fan;
13 (N) Evap. Core
14 (R) Heater Core;


                  F(inish)
    Select option:
                  M(ove a component)
```

Alterations to the sequence can be readily made. We have found that the approximate number of sequence re-orderings (r) will be

$$r = \frac{k^2 - 5k + 62}{1000}$$

where

$$k = \frac{\left(\frac{m_{MLT}}{m_{TOT}}\right)}{\left(\frac{c_{OFF}}{c_{TOT}}\right)} s$$

m(MLT) = number of multi-axis mates

c(OFF) = number of non-major axis components

Alterations made at this point are pull-out and re-insert elsewhere type (e.g. component number **11** must preceed number **4**). The Temperature Valve (S) must precede the Temperature Valve Actuator (F) while the Pipe Seal (J) must be assembled last. The manufacturing engineer for this product also established that the two bottom "boxes", (B) and (P), could be placed in the assembly system pallet and the case (L) could be snap-fit over both simultaneously. This means, along with realization that all the "side" components can be positioned and screwed to the case horizontally, that the product can be assembled with only **one** orientation on the pallet. This is a highly desirable condition and should always be sought. After completing this step, the sequence is printed below the exploded view drawing (see Figure 16).

ASSEMBLY PROCESS PLAN (Step P)

Having established an assembly sequence, we must now expand it into an assembly process plan. While many components can be assembled as a single task, there are cases where more than one task will be required (e.g. adhesive bonding, multiple threaded fasteners put in simultaneously). We may also need to re-orient the pallet (and all components already assembled), install and remove fixtures, as well as perform in-process tests.

Testing

The general assumption is that there will be a final test of the assembly after all components have been assembled. For various reasons, it might be desirable or necessary to have intermediate testing. Currently implemented is a user query about the number of such tests and the code name of the component(s) after whose assembly the test is to be performed as shown below:

```
          Assembly Planning Routine

        Current assembly sequence is :

   B MVH Sub-Assy.        J Pipe Seal
   P AI Case; AI Valv
   L Evaporator Case
   S Temperature Valv
   F Temp. Valve Actu
   O Solenoid  #1
   E Vacuum Element
   U Harness
   K Resistor Assy.
   T Solenoid Assy.
   H Motor & Fan;  Is
   N Evap. Core  Sub-
   R Heater Core; Hea
   A Cover


      A final test will always be included.
       Do you want other tests (Y or N) ?
```

For the ACM, we have specified one intermediate test, following the assembly of the wiring harness (U). Other testing (for more intricate products) may be desirable and can be implemented in the manner described.

Process Planner

If we state that it is highly advantageous to perform component assembly (whenever possible) in the vertical / down orientation only, it may be necessary to reorient whatever components have already been assembled in order to add a particular component. This probably requires fixtures which must be attached and removed as well as various rotators (pallet and possibly two trunnion). The resulting process plan will likely contain a large number of steps that have nothing (strictly speaking) to do with assembly.

Depending upon the product design, it may be possible to have only one orientation during the entire assembly (as we have seen for the Air Conditioning Module). SPM requires the user to make the choice between:

(**D**)ownward assembly only - product **must be** re-oriented so that every assembly task is straight down only. This is probably a requirement for any precision product. Our data indicates that the number of tasks (t) will be approximately

$$t_D = (1.5)^2 c = 2.25 c$$

(**F**)ixed orientation - product generally **will not be** re-oriented. This choice **allows** a total direction (180 deg.) switch between two consecutive components (e.g. Bottom to Top), resulting in the appropriate rotation being inserted into the plan. All components to be assembled in perpendicular directions require **no** re-orientation. We have found that the number of tasks will be approximately

$$t_F = (1.5) c$$

Figure 18 exhibits the **twenty-two** steps necessary to complete the assembly of the Air Conditioning Module's **fifteen** components. Because of the cognizant manufacturing engineer's cleverness, three of these non-assembly tasks can be eliminated in the next phase.

The fundamental procedure for the assembly of **each** component is:

1 Check the mate list to determine if the component mates with already assembled components.

2 If not, a fixture must be attached to the pallet before the current component can be assembled.(Note that the initial ACM assembly plan in Figure 18 prescribes adding a fixture in step 2 and removing it in step 13. These steps are not required for the final ACM plan.)

3 If so, and if there is only one mate, the conditions are obvious.

4 When multiple mates will occur, SPM assumes that the designer has ensured that no two (or more) **start** simultaneously. Thus, the planner determines only the most difficult mate and defines that as the required degree of difficulty. (This is an important problem in complex devices like automatic transmissions where multiple mates are common.)

5 It may be necessary to reorient in order to perform vertically down assembly. The rotations required will depend upon the orientation of the component in the final assembly and the current orientation of the already assembled components.

6 The assembly direction (T,L,P,R,F,B) of the present component may be opposite to that of the last already assembled component on that general axis (X, Y or Z). If so, all fixtures used for any prior component(s) on that axis are removed before assembly takes place. (Note task 13 in Figure 18.)

7 Certain mate types require multiple steps. Adhesive bonding usually requires three steps (apply adhesive, fixture components, oven cure). The other major multiple step mate type occurs when there is more than one fastener required to attach the current component; the planner interprets this as an insert bolts step followed by a torque bolts step.

The last two steps in every plan are final test and pack/unload assembly for shipment (which may may be as simple as unloading to the input buffer for the next level of assembly).

In addition to specifying appropriate actions for each step, the planner determines the following data (required to determine applicability of resources to each task):

ENVELOPE SIZE is determined by the overall dimensions of the product (Xmax, Ymax, Zmax). This has an important effect on the task time and size of equipment which will be needed.

RELATIVE ASSEMBLY DIFFICULTY (a **critical** assembly system design parameter) is found by taking the square root of the average degree of difficulty for the tasks which must be performed using a prescribed sequence. Figure 19 exhibits this information for several real products; we see that the relative assembly difficulty (A) can be approximated by

$$A = \frac{8.2 + \frac{S}{C}}{9}$$

Now that we have an assembly process plan, the next step is to create the task / resource matrix.

ESTABLISH THE TASK/RESOURCE MATRIX (Step M)

The final step in the SPM process is the creation of the task / resource matrix. Starting with an assembly process plan (described earlier), the tasks to be performed must be evaluated. The collection of such tasks will require some small set of resource **TYPES** (usually 7 or less), with the restriction that at least one type can perform each task. When a resource type is applicable to a task, we will specify an expected task time as well as the cost of the necessary hardware.

An experienced manufacturing engineer can readily make such judgements. A particular robot / end effector / part presentation method may be be desired. Certain types of test or heat-curing equipment could be necessary. Manual labor should usually be an option but may not be wanted in some cases. In general, out of the thousands of possibilities, very few resources are truly applicable to a specific product (or particular sub-assembly). Most of these decisions are based on the cumulative knowledge of the manufacturing engineer. The method described below combines our background performing just such work for various clients on a wide variety of products with the industrial data that we have been able to accumulate and uses it as a fundamental part of the SPM procedure.

Continuing the ACM example, we find by inspecting Figure 18 that two tasks (numbers 2 and 13) are not required. When the user eliminates them using the appropriate branch of the SPM program, we end up with the **twenty** tasks shown in Figure 20.

NOTE: All task time and hardware cost data derived here is intended to be a **guide**. While all test cases tried thus far have produced reasonable results, there is no guarantee of an absolute correspondence between this data and the precise data that is required for any particular assembly. The SPM program provides easy data alteration.

There will be two hardware cost categories, one for the resource and one for the "tooling". The latter will, in some cases, be called "station cost". Note that a Manual resource type has a resource cost of $200 while a Fixed Automation resource type has zero resource cost (all cost is in the station).

The assembly process plan (Figure 20) identifies all the activities necessary to perform complete assembly of a product or any meaningful portion thereof. Note that some of the tasks are not simply assembly.

Generally, the process plan (described earlier) consists of an ordered set of task descriptions. Since we seek a method for establishing information about task time and resource applicability, we need to know at least the following about each task (see Figure 20):

1. TYPE - the activity which is to take place.

2. MOTIONS REQUIRED - the type of movement (linear, planar, or spatial) necessary to get a component to its final position in the assembly.

3. LOAD - the estimated or actual weight of the component(s) or the force required for assembly (which may be due to driving torque).

4. DEGREE OF DIFFICULTY (B) - a measure of the complexity of the task.

5. TASK ACTIONS (N) - the number of activities which must take place during performance of the task. Note that some of the tasks shown in Figure 20 require more than one action.

Applicable Resource / Tool Specification

We start with the knowledge that there are three fundamental resource types; manual, fixed automation, and programmable automation. Each type has its own characteristics and may or may not be applicable to any given task. Part of the procedure defined here is to establish the applicability of each basic resource type to each task. We shall see that certain tasks can only be assigned to particular resources.

To establish a basis, cost and performance data from our consulting jobs has been plotted; linear, circular or simple hyperbolic curves (transposed where necessary) were fit as well as possible to that data. Recall that we are not attempting to provide absolute data, but only a very good first approximation to the necessary information. Application to particular products may allow an increase in the accuracy.

We have found three basic cost contributing parameters; load (altered by a safety factor), reach (altered by another safety factor), and Degrees of Freedom required. In general, we can express the programmable resource cost, the manual "tooling" cost, or the fixed automation "station" cost as:

$$H = f_P H_P + f_D H_D + f_R H_R$$

where the f parameters are decimal (and must sum to 1.0) and the H parameters are defined for each resource type (subsequently). The P subscript denotes load (either weight <u>or</u> force) requirement. Degrees-of-freedom needed are symbolized by the D subscript, while the R subscript designates reach requirement.

Basic Characteristics

First we need to specify that load and reach requirements have a (user determined) safety factor applied according to:

$$P = s_P P'$$

for load (P'), and

$$R = s_R R'$$

for reach (R').

Typical values are: s(p) = 1.5 s(r) = 1.2

Note that weight or force is to be derived from the product knowledge base. Reach requirement is currently derived from the final assembled product envelope; we assume that no part will have to be moved farther than from part presentation location to the middle of the product. The nominal move distance is:

$$R' = \sqrt{X_{max}^2 + Y_{max}^2 + Z_{max}^2}$$

where only the appropriate X, Y and/or Z (specified by the task motion requirement) are used.

Cost component attributable to degree(s)-of-freedom :

We define the degrees-of-freedom requirement as follows: for linear motion (X or Y or Z), D = 4; for planar motion (X-Y or X-Z or Y-Z), D = 5; for spatial motion (X-Y-Z), D=6. Using our historical data as a guide, we have derived the following approximations for hardware cost as a function of degrees-of-freedom:

Manual "Tooling" Cost, (k\$) :

$$H_D = \left[\frac{D-1}{2}\right]$$

Fixed Automation "Station" Cost, (k\$) :

$$H_D = 35\,D^2$$

Programmable Resource Cost, (k\$) :

$$H_D = 200 - 17.5\,D + 17.5\sqrt{D^2 - 27\,D + 127}$$

The weighting factor f(D) for degree(s)-of-freedom (for each resource type) will be based upon the requirements for all the tasks that each type can perform on the presently considered product (or sub-assembly). It must be determined for every pass through the SPM procedure.

Cost component attributable to load (weight or force) to be accommodated

We have derived the following cost vs. load characteristic approximations from our historical data:

Manual "Tooling" cost, (k$) :

$$H_P = 2$$

Fixed Automation "Station" cost, (k$) :

$$H_P = 30 + 14\,P$$

Programmable Resource cost, (k$) :

$$H_P = -10 - 1.7\,P + 2\sqrt{P^2 + 140\,P + 25}$$

The weighting factor f(P) for load (for each resource type) will be based upon the requirements for **all** the tasks that each type can perform on the presently considered product (or sub-assembly). It must be determined for every pass through the SPM procedure.

Cost component attributable to reach requirement

As described earlier, the reach necessary is derived from the task motion requirement as it relates to the overall envelope. We have determined the following cost vs. reach requirements:

Manual "Tooling" cost, (k$) :

$$H_R = 307.92 - \sqrt{93893.68 - 4.76\,R - R^2}$$

Fixed Automation "Station" cost, (k$) :

$$H_R = -220 + 19.2\,R + 19.2\sqrt{R^2 - 23.2\,R + 149.2}$$

Programmable Automation cost, (k$) :

$$H_R = 1450 + 17.5\,R - 17.5\sqrt{R^2 + 140\,R + 6750}$$

The weighting factor f(R) for reach requirement (for each resource type) will be based upon the requirements for **all** the tasks that each type can perform on the presently considered product (or sub-assembly). It must be determined for every pass through the SPM procedure.

Cost Weighting Factors

At first it was felt that these factors would have to be user specified. After running a few examples, we found that the costs could be substantially different for various weighting allocations. Each product/sub-assembly will have a collection of requirements which must be translated into representative cost characteristics. The cost of performing any task is thus related to the cost behavior of the total tasks to be accomplished. Choosing a serial relationship for these cost factors would result in the highest cost contributor having the largest weighting factor. We prefer to allocate the cost contributions more evenly and therefore use a parallel relationship. Since consistent results are needed, the following method for determining these factors has been instituted.

Going through the various tasks, determine the "un-weighted" costs as shown above for each resource type; keep track of the total H(d), H(p), and H(r) values. When all tasks have been looked at, find the nominal total cost from:

$$H_{Total} = \frac{1}{\frac{1}{H_{D_{Total}}} + \frac{1}{H_{P_{Total}}} + \frac{1}{H_{R_{Total}}}}$$

which can be rearranged as:

$$1 = \frac{H_{Total}}{H_{D_{Total}}} + \frac{H_{Total}}{H_{P_{Total}}} + \frac{H_{Total}}{H_{R_{Total}}} = f_D + f_P + f_R$$

The cost weighting factors are then readily determined from:

$$f_D = \frac{H_{Total}}{H_{D_{Total}}}$$

$$f_P = \frac{H_{Total}}{H_{P_{Total}}}$$

$$f_R = \frac{H_{Total}}{H_{R_{Total}}}$$

This method has produced very reasonable cost weighting factors for all examples tried thus far. Average values for those cases are:

	Manual	Programmable	Fixed
Load factor	0.314	0.343	0.160
Reach factor	0.308	0.312	0.369
DoF factor	0.378	0.345	0.478

These characteristics are not fixed! Each data set which is processed will have its own particular values. Note that Figure 21 (top) exhibits quite different values.

We now have all the parameters for defining some of the basic resource and tool costs. The Programmable "Tooling" cost will be determined a bit later.

Actual Hardware Cost

The calculations above assumed a straight-forward task. We adjust the cost based on the individual task degree-of-difficulty parameter (B) - note typical data in Figure 20. Using our historical data as a basis, we define hardware cost to be:

$$C_H = H\sqrt{B}$$

This can be expressed as (k$) :

$$C_H = \sqrt{B}\left[f_P H_P + f_D H_D + f_R H_R\right]$$

To minimize numerical complexity, C(H) values are rounded to the next highest $0.1k for a Manual resource and to the next highest $0.5k for any automated resource. Figure 21 exhibits the "raw" task / resource matrix data. Note that these C(H) costs apply to "Tooling Cost" for manual and fixed automation resources and apply to "Resource Cost" for programmable resources. Manual resource hardware cost is specified to be $0.2k (workbench and chair) while fixed automation resource cost is zero (the cost is all in the specially designed station).

Programmable "Tooling" cost

The gripper and material presentation/handling cost required by a particular task when using a programmable resource is defined to be a function of the resource cost and the degree of difficulty expected. Using our historical data as a guide, we express this cost as (k$) :

$$C_T = 3\sqrt{B} + \frac{C_{H_{Prog}}}{8}$$

Examples of all calculated costs (rounded to the next \$0.5k for automated resources and to the next \$0.1k for manual resources) can be readily observed in Figure 21. Note the wide variety of Programmable resource hardware costs; we shall reduce the variety to a representative few later.

"Tool" Number

Each task that can be performed by a Fixed Automation resource has an individual "tool" number; therefore, only one task can (generally) be performed at a Fixed Automation station. Manual and Programmable resources, however, have the potential to use the same "tool" for more than one task. When that condition is observed by a designer/engineer (who may have designed the "tool" for just such a case), the same "tool" number and cost are assigned to each of the tasks. In the present case, we have task characteristics and an estimated "tool" cost; our goal is to determine automatically whether any tasks with equivalent cost are similar enough to use the same "tool".

We start assigning "tool" numbers at 1 (for each of the three generic resource types) for the first applicable task. As we go through all other tasks, a check is made to see whether the "tooling" cost of the task presently being investigated is exactly the same (recall the rounding-up procedure) as any previously established. If so, then the following parameters are checked to determine compatibility:

Degree of Difficulty
Motion requirement similarity
Weight in same order of magnitude
Task type

When all audits agree, the same "tool" number is assigned. If any one check does not agree, a new "tool" number is specified. Figure 21 readily exhibits this behavior. This scheme may specify that two tasks require the same tool but in actuality a common tool is not desired. There are two ways to alter this condition. Within SPM, the user can modify the process plan data for either task. The second (and much easier) method is to alter one tool number in the input data used for the assembly system synthesis program.

Task Time

Earlier, we stated that task time is a function of reach. It also is directly related to the degree of difficulty of a task, the number of activities (N) that must take place within the task, and the relative assembly difficulty (A). We will determine the "fundamental" task time for a Manual resource and let the user specify (by type, u(f) and u(p)) what portion of that time is required when the task is performed by Fixed Automation or Programmable Automation.

Manual time, t(M) (seconds)

$$t_M = (t')^A$$

where, based on our historical data, we estimate t' as:

$$t' = \left\{.036\left[(21 - 5B)(3 + R') + 40B^2 - 90(B - 1)\right]\right\} N^{.3333}$$

It is important to realize the significance of the relative assembly difficulty (A). The minimum value is 1.0 while the largest usable value is about 1.5 for the cases tested thus far.

Fixed Automation time, t(F) (seconds)

$$t_F = u_F t_M$$

The relative time factor u(f) has the default value 0.3333 but can be changed by the user.

Programmable Automation time, t(P) (seconds)

$$t_P = u_P \, t_M$$

The default value for u(p) is 0.900 which can be changed by the user as required.

Figure 21 shows the variety of task times which occur in a typical assembly problem.

Resource Type Applicability

Note that some tasks can not be performed by all generic resource types, but that at least one type must always be applicable. For the ACM example (Figure 21), tasks 8 and 18 require manual labor; these are both very difficult tasks (see Figure 20). Task 19 is specified to be an automated (only) test. The general tests for applicability of a resource type to a task involve the type of task, the degree-of-difficulty involved, and the degrees-of-freedom required. Certain task types can not be performed by some resources:

(1). Manual can not perform "O" (oven), "V" (automated test), or "R" (pallet/trunnion rotator).

(2). Fixed Automation can not perform any task with a degree-of-difficulty greater than 2, a three-dimensional motion requirement (unless it is a "V" task), or if it is rework.

(3). An appropriate level of programmable automation can be used for all tasks except "M" (measurement), "G" (selective fit), "V" (automated test), "O" (oven bake), "Q" (manual rework), or "R" (pallet/trunnion rotate).

Selecting Representative Programmable Resources

For the example shown in Figure 21, we can see that the range of calculated programmable resource hardware costs is \$ 27.5 k to \$88.5 k. This data is bar charted in Figure 22. We seek a means to reduce this variety to a few resource costs which will represent the requirements for the assembly system design problem. We use a heuristic procedure based on the observation that programmable automation is seldom economical unless multiple tasks can be performed. The method requires ranking in ascending order and counting (C) the potentially applicable tasks (note that 8, 9, 18 and 19 can not be performed by programmable resources). The maximum number of programmable resource types, n(p), that can be used (based on the industrial practice of minimizing system

complexity) is determined from the following:

$C <= 5$	$n(p) = 1$
$5 < C <= 12$	$n(p) = 2$
$13 < C <= 24$	$n(p) = 3$
$25 < C$	$n(p) = 4$

By stepping through the calculated resource costs (rounded up to the next $5.0 k), we can separate the collection into groups. We specify which of these cost categories is to be used by first determining the incremental tasks that can be performed. Starting with the lowest cost programmable resource, we establish the number of consecutive tasks that are assignable (normally, they occur in groups of sequential tasks). For example, Figure 23 shows that the ACM assembly needs a $30k device that can do four tasks (in blocks [10,11] and [15,16] - see Figure 22) while the $35k device can do four additional tasks (in blocks [6,7], [10,11,12,13] and [15,16]). Note that task 3 will not be counted until the cost increases to $50k but that task 14 gets included immediately at $40k since it is the maximum of a lesser cost block [10,11,12,13,14,15,16].

For this example, we have C=18, n(p)=3, and the programmable resource cost categories of $30k, $35k and $50k. They are shown as the horizontal lines seen in Figure 22. While this scheme errs on the high cost side, it appears to be reasonable given that virtually all the calculations are based on *approximations* (which are curve fits to historical data).

Programmable Resource Cost / Performance Data

Each Programmable resource (determined as shown in the last section) has identifiable cost and performance parameters. If C(H) is the hardware cost (k$), we can estimate the following:

rho Factor *(total investment required)/(hardware cost)*

$$\text{rho} = 1 + \frac{C_H}{40}$$

Up-time Expected

$$e = (0.95)^A$$

Recall that (A) is the relative assembly difficulty.

Operating / Maintenance Rate ($/hr)

$$O_H = \frac{C_H}{50}$$

Tool Change Time (seconds)

$$T_C = 2.5 + \frac{C_H}{20}$$

Maximum Stations per Worker

$$m_S = \frac{1600}{C_H + 150} - 3$$

The user has the option of changing the hardware cost for any of the programmable resources found by SPM. While not usually desirable, specific conditions may require such action. Whenever the resource hardware cost is altered, all of the above parameters are also changed automatically. SPM allows the user to change the value of any parameter whenever circumstances make it necessary.

Manual and fixed automation resource types are much easier to specify. We use the following values:

Manual Resource cost /performance data.

Rho factor = 1.2

Up-time expected = 100% times (.90 raised to the A power)

Operating / maintenance rate = .5

Tool change time = 2.0

Maximum stations per worker = 0.833

Fixed Automation Resource cost /performance data.

Rho factor = 1.0

Up-time expected = 100% times (.95 raised to the A power)

Operating / maintenance rate = A (numerical approximation)

Tool change time = 0

Maximum stations per worker = 1 + [120 times applicable tasks divided by F.A. total cost]

Assembly System Design Input Data

We now have all the task / resource information that is required for an assembly system design program (such as A.S.D.P.). Figure 24 displays the information derived by the methods shown here; it is presented in a format somewhat familiar to A.S.D.P. users. Note that some tasks can not be performed by some of the resources; the more expensive the Programmable resource, the more tasks a device in that cost category can perform. Some "raw" programmable resource data (tasks 17 & 20 in Fig.21) does not appear in Figure 24 since the device required is more expensive than any of those specified.

The combinatorics available for comparison in a system design problem is totally dependent upon the working days per year, the shifts available per day, and the required annual production volume.

With the limitation of consecutive task assignments, we can observe in Figure 24 that the maximum possible tasks at any station is eighteen (applicability depends on whether station time and/or space is available), while the minimum is, of course, one.

SUMMARY

We can estimate the likelihood of establishing a cost-effective assembly system by determining the average number of applicable resource types for each task. Note that, in general, some tasks can be performed by only one resource type; we call those unique tasks. Let N = the number of general tasks = total tasks - unique tasks (for the ACM, this value is 20-4 =16). If we take the Nth root of the total task-resource combinations possible (the product of the applicable resource types for each task; for the ACM, this value is 207 million), we obtain an applicable resource types per task number as exhibited in the following statistics:

Example	Total tasks	General tasks	Resource types	Appl. res. per task
Generic Gimbal	24	8	6	3.8
Large Trans. Final Assy.	26	6	6	3.9
Seeker Head Assembly	39	11	8	4.1
Seeker Head - A	17	5	6	3.1
Seeker Head - B	24	8	6	3.6
Complete A. C. M.	20	16	5	3.3
Optics Sub-assembly	19	3	5	3.2
Simplified TRAX Final Assy.	19	7	6	3.6
Trans-axle Final Assy.	35	8	7	3.9
Disk Head Assembly	30	5	7	3.5
Solid Rear Axle Assembly	28	3	7	4.1

It is interesting to note that these data sets exhibit approximately the same number of applicable resource types for each non-unique task. The nominal relationship is approximated by

$$AART = 2.7 + \frac{t_{TOT}}{28}$$

If the actual average applicable resource types (AART) per task is less than the value calculated, it is desirable (but not always possible) to simplify the product design such that more equipment choices are available. Seeker Head - A and Disk Head Assembly are both candidates for such action.

We generally believe that the larger the number of resource alternatives (for tasks and also usually for clusters of sequential tasks), the more cost-effective the resulting assembly system will be.

CONCLUSION

A method for making the connection between product data and the requisite assembly system design input data has been described. Using physical information about a product (or portion thereof) and including component mate characteristics, we are now able to

develop the assembly SEQUENCE,

establish the assembly PROCESS PLAN,

and create the task / resource MATRIX.

Relationships between various parameters have been shown. The product engineer/designer as well as the manufacturing engineer now have a means for evaluating the effects of product characteristics on the system needed to assemble it.

The techniques shown have been implemented in an IBM PC (or compatible) program called SPM. Connections to ASDP (Ref. 1) are routinely made. They are being successfully applied to a variety of commercial, industrial and military products.

<u>REFERENCES</u>

1. *A.S.D.P. (Assembly System Design Program)*, Copyright 1985, The Charles Stark Draper Laboratory, Inc., Cambridge, MA. Version 6.0, 1989.

2. Andreasen, M.M., et.al., *Design for Assembly*, Springer-Verlag, 1983.

3.. Boothroyd, G., C. Poli and L. Murch, *Automatic Assembly*, New York, Marcel Dekker, 1983.

4. Huthwaite, Bart, "Checklist for DFM," *Machine Design*, Jan. 25, 1990.

5. Lewis, G.M. (Ed.), *Design for Assembly and Automation*, Xerox Automation Institute, Webster, NY, 1985.

6. Sturges, R.M., Jr., "A Quantification of Manual Dexterity: The Design for an Assembly Calculator," *Robotics & Computer-Integrated Manufacturing*, Vol. 6, No. 3, pp. 237-252, 1989.

7. Baldwin, D.F., T.E. Abell, M.C. Lui, T.L. De Fazio and D.E. Whitney, "AN Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products," submitted to *IEEE Journal of Robotics & Automation,* March 1990.

8. Bourjault, A., *Contribution a une approche methodologique de l'assemblage automatise ; elaboration automatique des sequences operatoires*, These d'Etat, Universite de Besancon Franche-Comte, 1984.

9. De Fazio, T.L. and D.E. Whitney,"Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, RA-3(6):640-658, December 1987.

10. Delchambre, A., P. Gaspart and A. Wafflard, "An Automatic Assembly Planning Approach," *10th International Conference on Assembly Automation*, Kanazawa, Japan, 23-25 October 1989.

11. Homem de Mello, L.S. and A.C. Sanderson, "Automatic Generation of Mechanical Assembly Sequences," The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1988.

12. Homem de Mello, L.S., *Task Sequence Planning for Robotic Assembly*, PhD. Thesis, Carnegie Mellon University, Pittsburgh, PA, 1989.

13. Homem de Mello, L.S. and A.C. Sanderson, "Evaluation and Selection of Assembly Plans," *1990 IEEE International Conference on Robotics and Automation*,

14. Hoummady, A. and K. Ghosh, "Generation and Evaluation of Assembly Sequences in Computer-Automated Process Planning," *International Journal of Computer Applications in Technology*, Vol. 2, No. 3, pp. 151-158.

15. Ko, H. and K. Lee, "Automatic Assembling Procedure Generation from Mating Conditions," *Computer-Aided Design*, Vol. 19, No. 1, pp. 3-10, Jan/Feb 1987.

16. Kroll, E., E. Lenz and J.R. Wolberg, "A Knowledge-Based Solution to the Design-for-Assembly Problem," *Manufacturing Review*, Vol. 1, No. 2, pp. 104-108, June 1988.

17. Miller, J.M. and R.L. Hoffman, "Automatic Assembly Planning with Fasteners," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May 14-18, 1989.

18. Takeyama, H., et.al., "Study on Automatic Determination of Assembly Sequence," *Annals of the CIRP*, Vol. 32, No. 1, pp. 371-374, Harrogate, England, August, 1983.

19. Wilson, R.H. and J-F Rit, "Maintaining Geometric Dependencies in an Assembly Planner," *1990 IEEE International Conference on Robotics and Automation*,

A view of CONCURRENT DESIGN

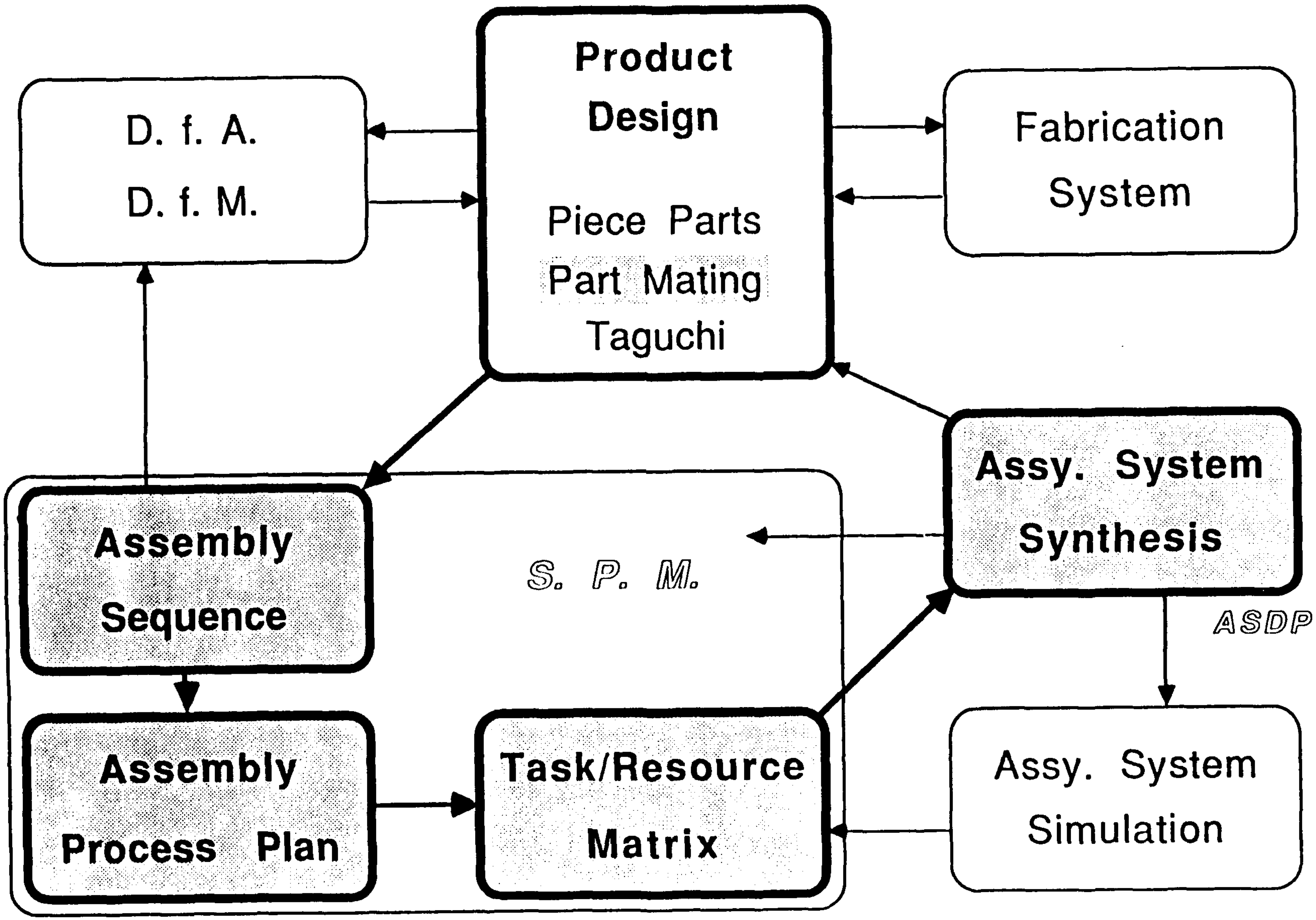
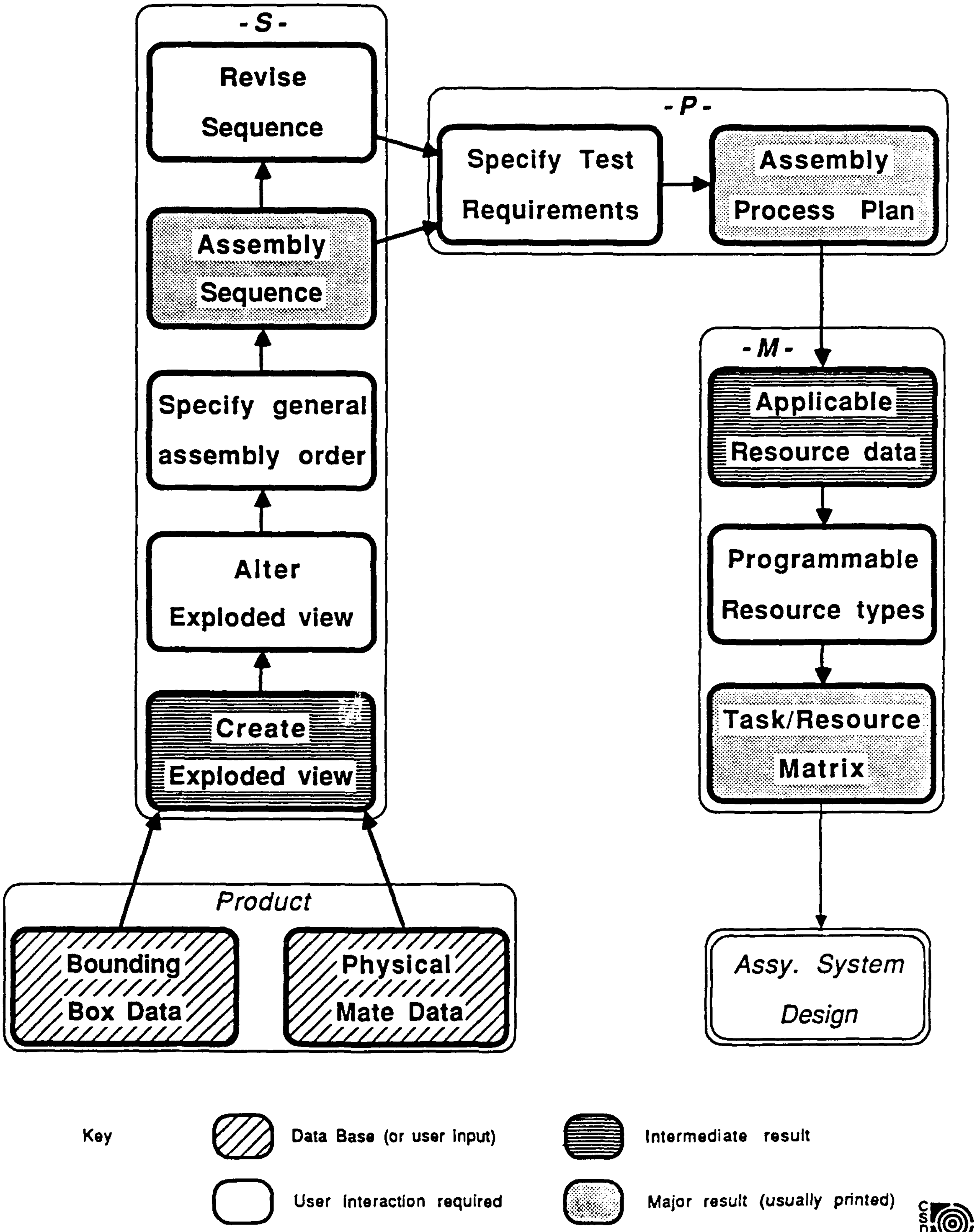


Figure 1

S. P. M.



REG
21 March 1990

Figure 2

Air Conditioning Module

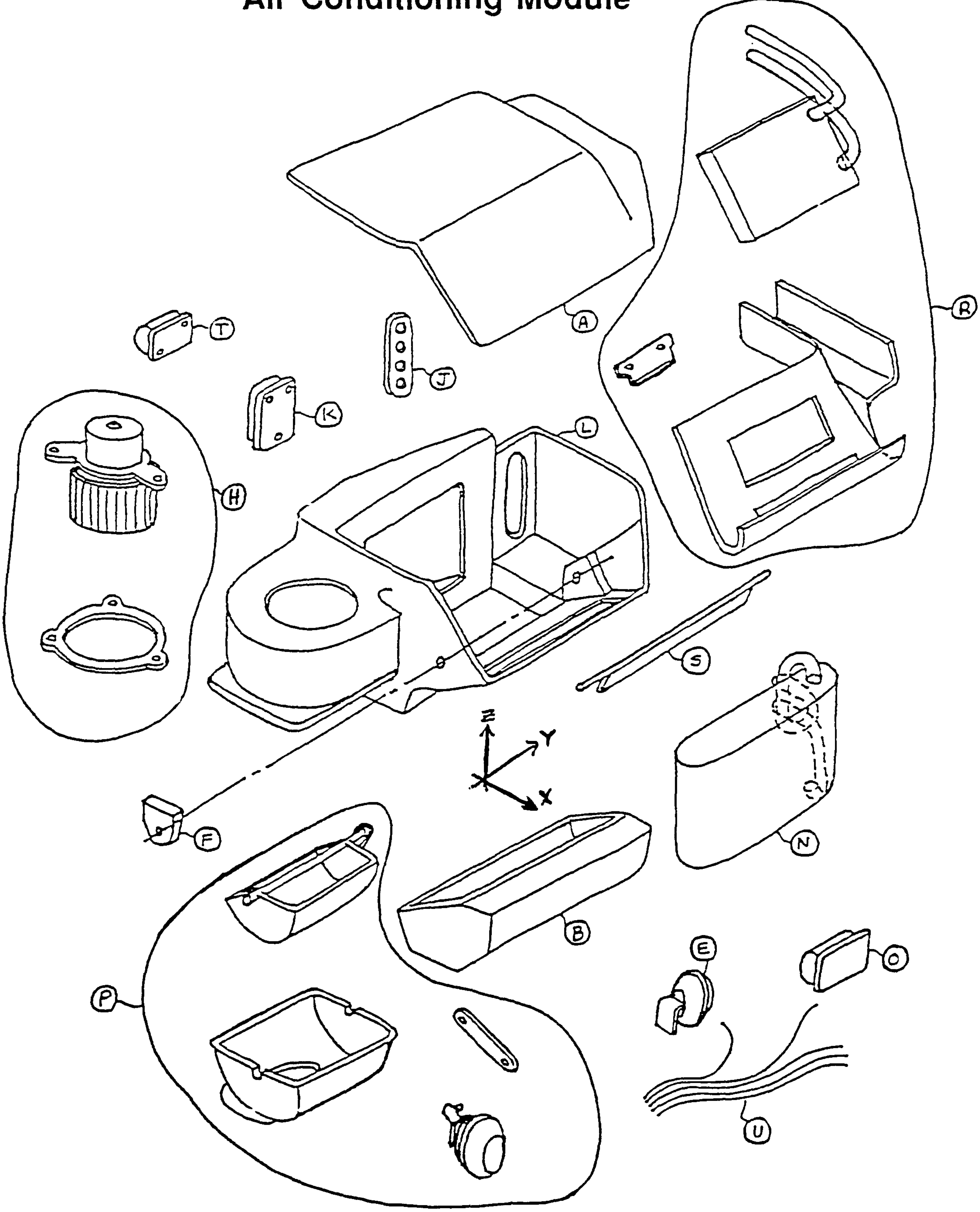


Figure 3

Air Conditioning Module

COMPDATA

Total Assembly Bounding Box (x,y,z).			Number of Components
15.6	23.0	11.5	15

Component Bounding Planes

Code	Wt	Xmin	Xmax	Ymin	Ymax	Zmin	Zmax	Description
A	1	2.2	12.7	0	16.1	4.8	11.5	Cover
B	4	6.4	14.3	0.5	12.6	0	4.8	MVH Sub-Assy.
E	1	6.0	8.4	-2.5	0	8.3	10.8	Vacuum Element #2
F	2	5.7	8.7	-0.8	0	3.6	5.6	Temp. Valve Actuator
H	4	1.5	5.7	-6.0	-1.8	5.3	10.8	Motor & Fan; Isolator
J	1	2.0	2.3	14.0	15.3	3.8	9.8	Pipe Seal
K	1	-1.2	0	1.7	3.7	8.4	10.9	Resistor Assy.
L	5	0	12.6	-6.9	16.2	0	11.2	Evaporator Case
N	4	2.6	6.8	1.2	15.0	0.4	10.0	Evap. Core Sub-Assy.
O	1	7.8	10.2	-1.4	0	6.3	7.8	Solenoid #1
P	4	0.7	7.2	-6.6	-0.6	1.0	5.0	AI Case; AI Valve; Vacuum Element; Link
R	7	4.8	10.2	3.0	15.0	3.6	10.9	Heater Core Shroud; Heater Core; Clamp
S	1	6.5	7.5	0.9	10.9	2.7	5.2	Temperature Valve
T	1	-1.2	0	1.7	3.7	4.0	6.0	Solenoid Assy.
U	1	-0.3	0	-10.0	14.0	0	0.3	Harness

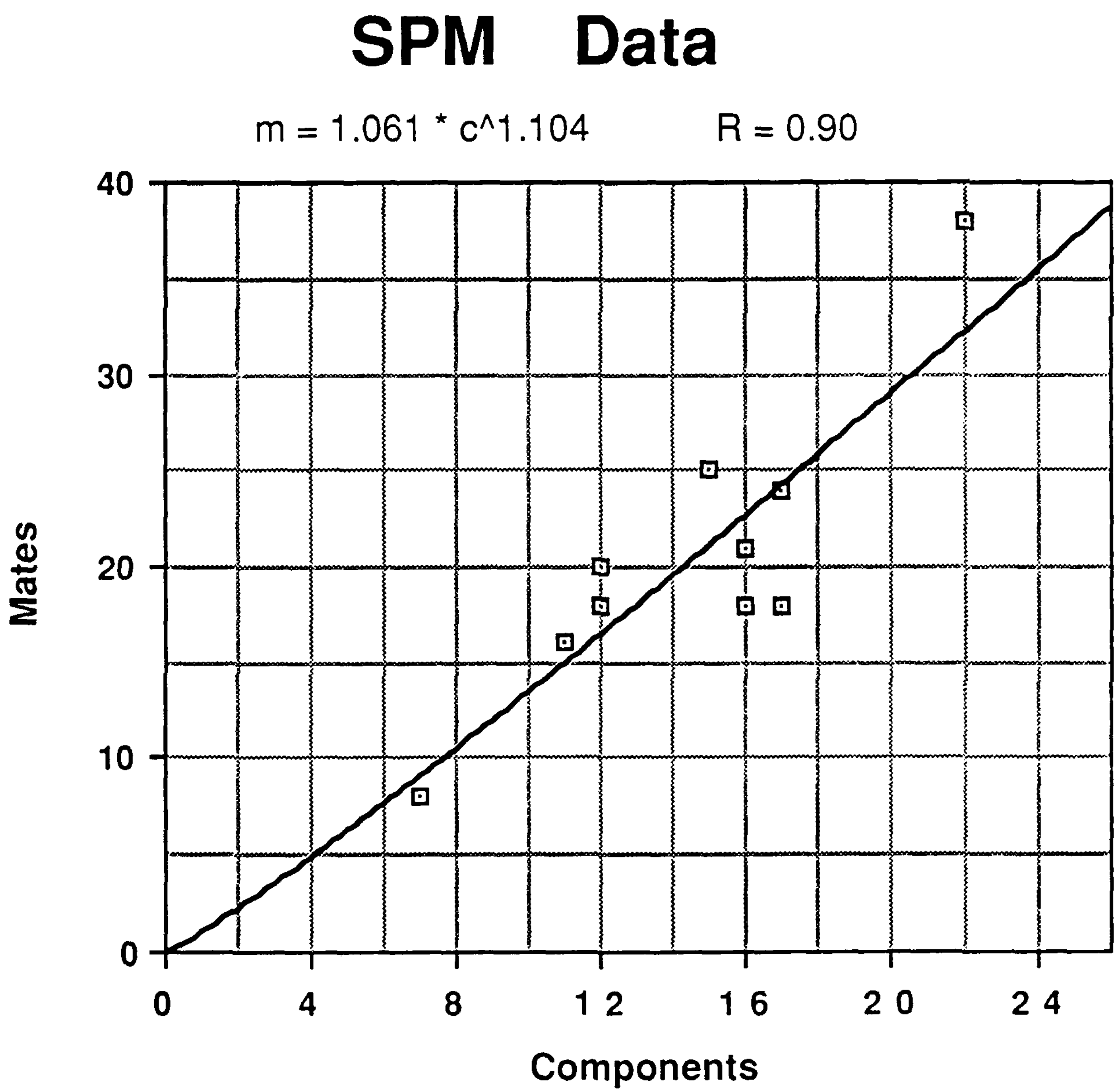
Figure 4

Air Conditioning Module MATEDATA - Input

Total Number of Mates 26

Codes	type	number	Fit	Direction
A, L	screw(s)	7	n/a	Z
A, R	placement	1	n/a	Z
B, L	snap-fit	1	n/a	Z
E, L	snap-fit	1	n/a	Y
F, L	screw(s)	2	n/a	Y
F, S	press-fit	1	transition	Y
H, L	placement	1	n/a	Z
J, L	press-fit	1	n/a	X
J, R	press-fit	1	n/a	X
J, N	press-fit	1	n/a	X
K, L	screw(s)	3	n/a	Y
N, L	*placement*	1	n/a	Z
N, R	placement	1	n/a	Z
O, L	screw(s)	2	n/a	Y
P, L	snap-fit	1	n/a	Z
R, L	placement	1	n/a	Z
S, L	bearing race / bushing	2	transition	Z
T, L	screw(s)	2	n/a	X
U, P	snap-fit	1	n/a	X
U, E	snap-fit	1	n/a	X
U, K	snap-fit	1	n/a	Y
U, O	snap-fit	1	n/a	X
U, T	snap-fit	1	n/a	Y
U, F	snap-fit	1	n/a	X
U, H	snap-fit	1	n/a	X
U, B	snap-fit	2	n/a	X

Figure 5



SPM Data
m = 1.061 * c^1.104
R = 0.90
40
30
20
10
0
Mates
0
4
8
1 2
1 6
2 0
2 4
Components

Figure 6

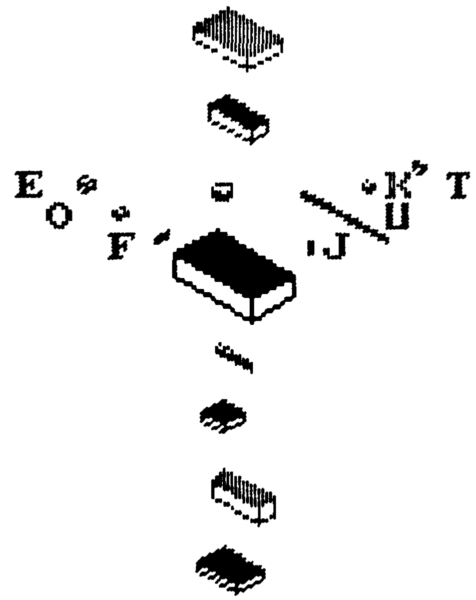
Air Conditioning Module MATEDATA - Processed

Total Number of Mates 26

Codes	type	DoD	DoF	Multiple	Sub-tasks	Direction
A, L	B	1	1	7	1	Z
A, R	P	1	3	1	1	Z
B, L	E	1	1	1	1	Z
E, L	E	1	1	1	1	Y
F, L	T	2	1	2	1	Y
F, S	L	3	1	1	1	Y
H, L	P	1	3	1	1	Z
J, L	L	3	1	1	1	X
J, R	L	3	1	1	1	X
J, N	L	3	1	1	1	X
K, L	T	1	1	3	1	X
N, L	P	1	1	1	1	Z
N, R	P	1	3	1	1	Z
Q, L	T	1	1	2	1	Y
P, L	E	1	1	1	1	Z
R, L	P	1	3	1	1	Z
S, L	A	3	1	1	1	Z
T, L	T	1	1	2	1	X
U, P	E	2	1	1	1	X
U, E	E	2	1	1	1	X
U, K	E	2	1	1	1	Y
U, O	E	2	1	1	1	X
U, T	E	2	1	1	1	Y
U, F	E	2	1	1	1	X
U, H	E	2	1	1	1	X
U, B	E	2	1	1	2	X

Figure 7

AIR CONDITIONING MODULE Final Assy.



(A) Cover

(R) Heater Core; He

(H) Motor & Fan; I

(L) Evaporator Case

(S) Temperature Val

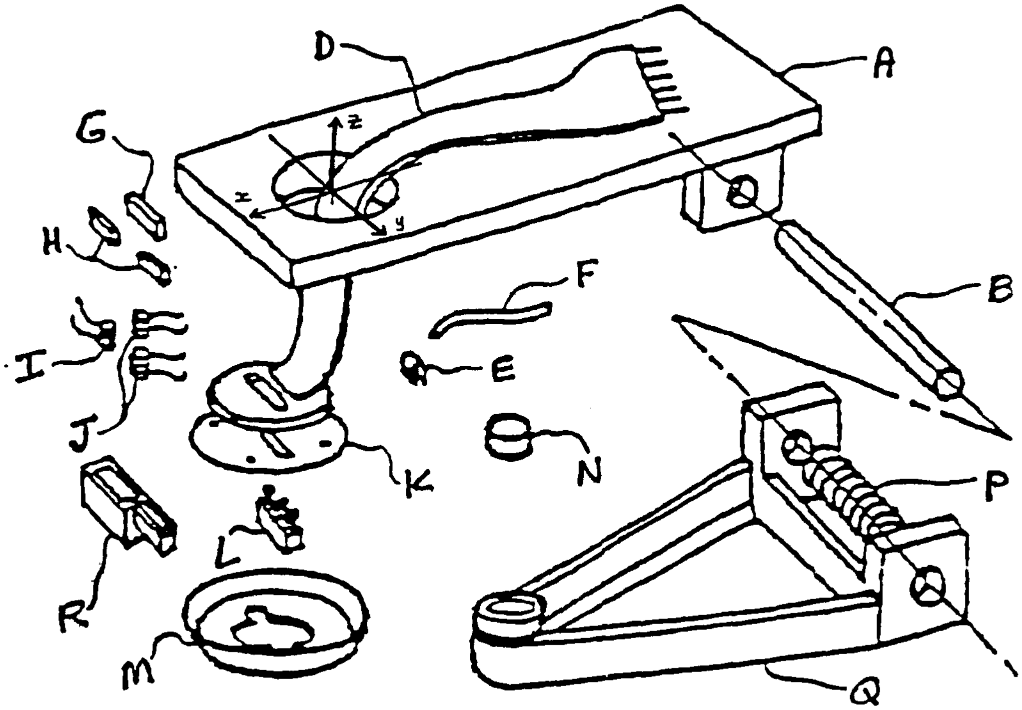
(P) AI Case; AI Val

(N) Evap. Core Sub

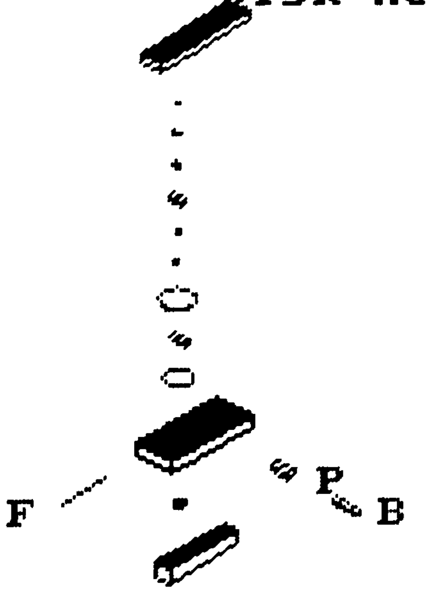
(B) MVH Sub-Assy.

(M)ove component (C)omplete

Figure 8



Disk Head Assembly



(D) Flex Circuit

(E) Clip
(H) Back Bars (2)
(G) Butt Bar
(R) Positioner Tray
(J) R/W Coils (2)
(I) Erase Coil
(M) Shield

(L) Slider
(K) Diaphragm

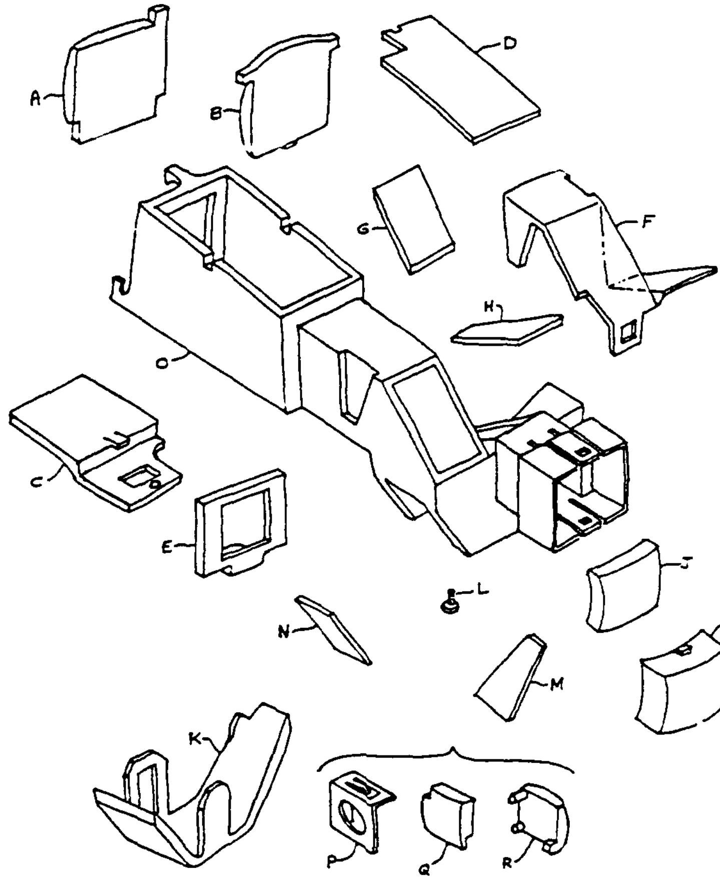
(A) Base Plate

(N) Pad

(Q) Arm

(M)ove component (C)omplete

Figure 9



OPTICS Sub-assembly

J
I

(D) Upper Lens COVE
(B) Field LENS (#2)
(E) Picture FRAMER

(G) Rectangular MIR

(O) CASE

(C) Lower Lens COVE
(M) Trapezoid MIRRO

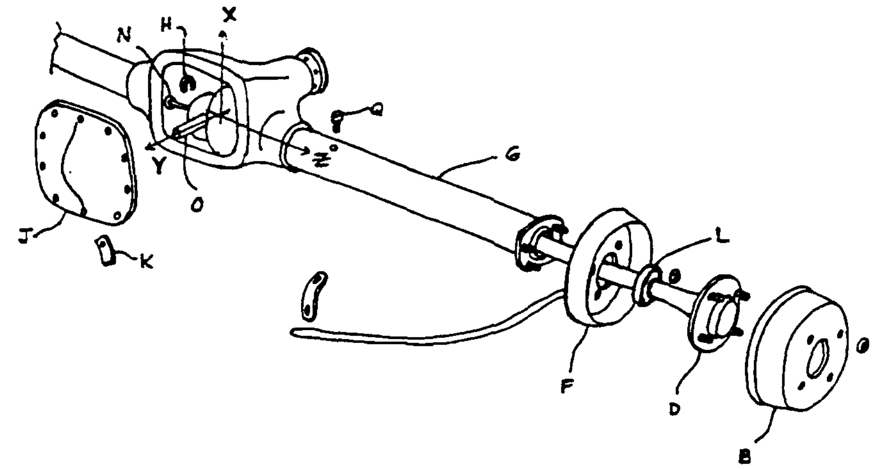
(H) Trapezoid MIRRO
(F) Upper Mirror CO

(P) Spring BRACKET;
(N) Rectangular MIR

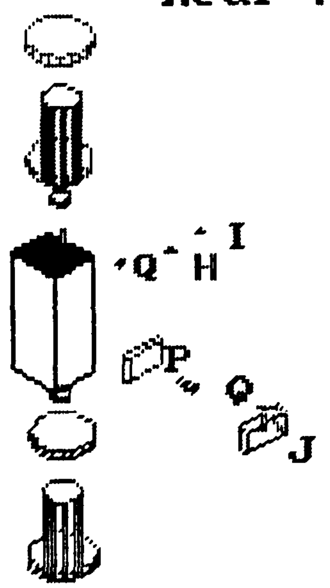
(K) Lower Mirror CO
(L) Adjusting SCREW

(M)ove component (C)omplete

Figure 10



Rear Axle Assembly



(B) Brake Drum (RH)

(D) Rear Axle (RH)

(F) Brake Assy. (RH
(M) Outer Bearing;
(N) SLB

(G) Axle Assy.

(L) Outer Bearing;

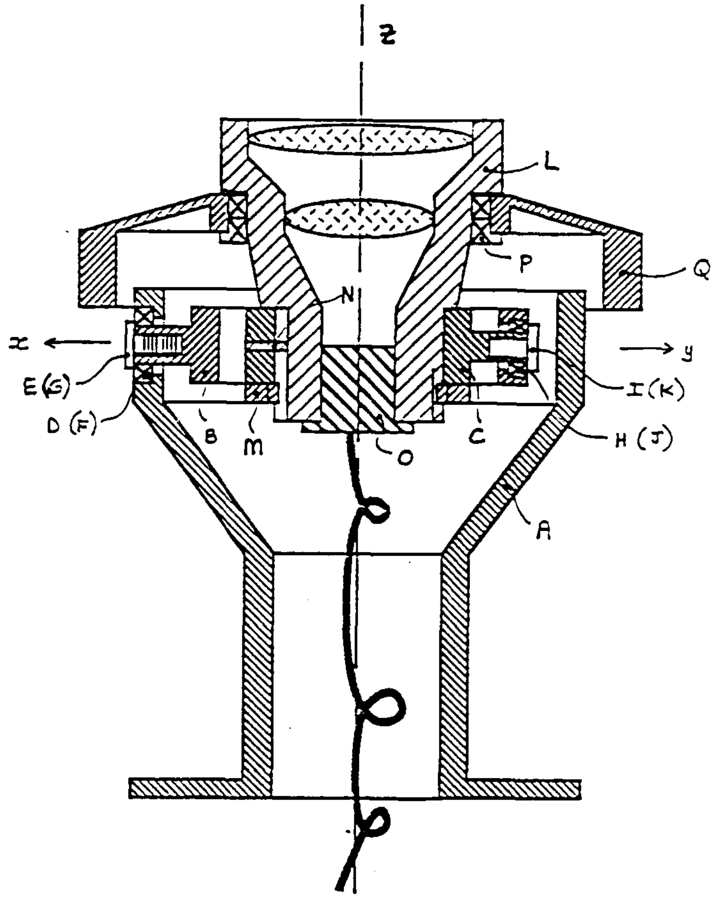
(E) Brake Assy. (LH

(C) Rear Axle (LH)

(A) Brake Drum (LH)

(M)ove component (C)omplete

Figure 11


z
L
P
Q
N
x
y
E(G)
D(F)
B
M
O
C
I(K)
H(J)
A

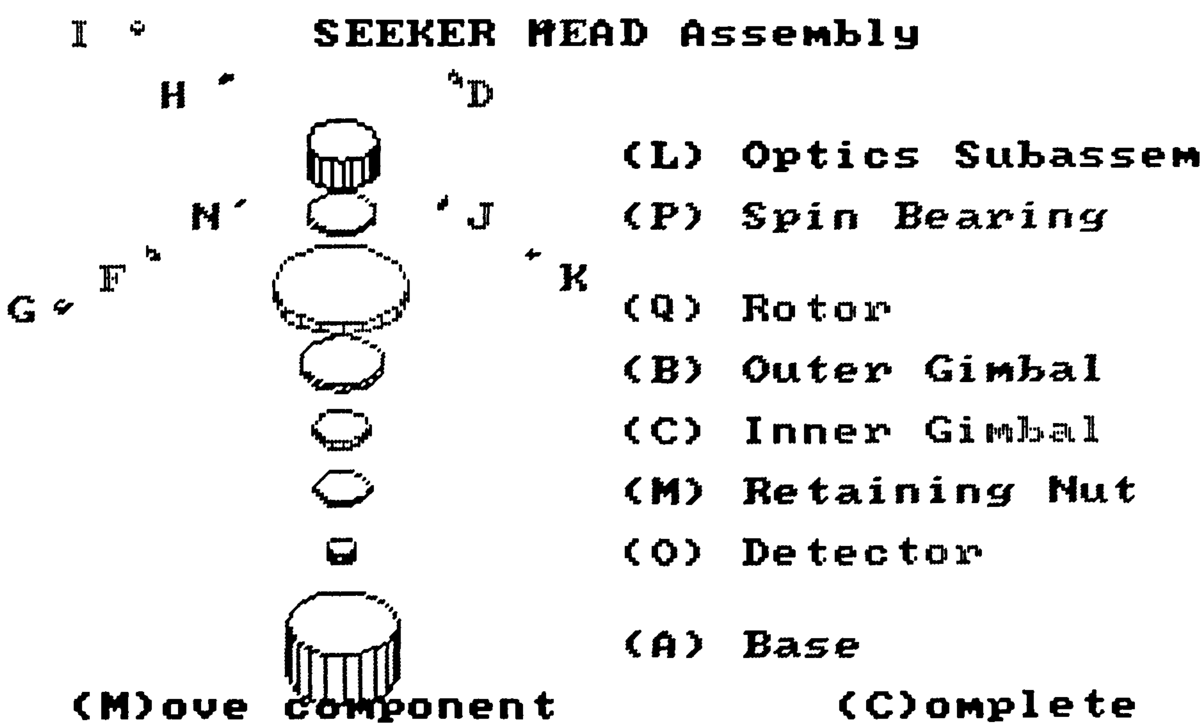
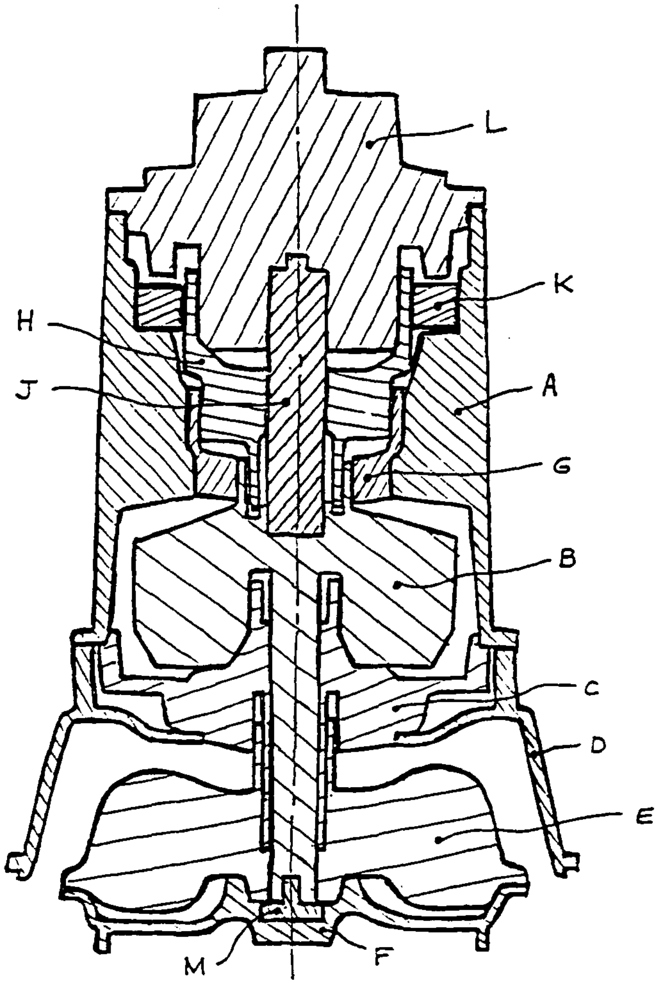
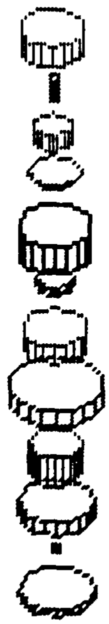

SEEKER HEAD Assembly
I
H
D
N
J
F
K
G
(L) Optics Subassem
(P) Spin Bearing
(Q) Rotor
(B) Outer Gimbal
(C) Inner Gimbal
(M) Retaining Nut
(O) Detector
(A) Base
(M)ove component
(C)omplete

Figure 12



LARGE TRANSMISSION Final Assy.



(L) REAR HOUSING -

(J) MAIN SHAFT
(H) P2 CARRIER - P3

(K) CLUTCH STACK
(A) MAIN HOUSING -

(G) P1 CARRIER - P2

(C) OIL PUMP ASSY.

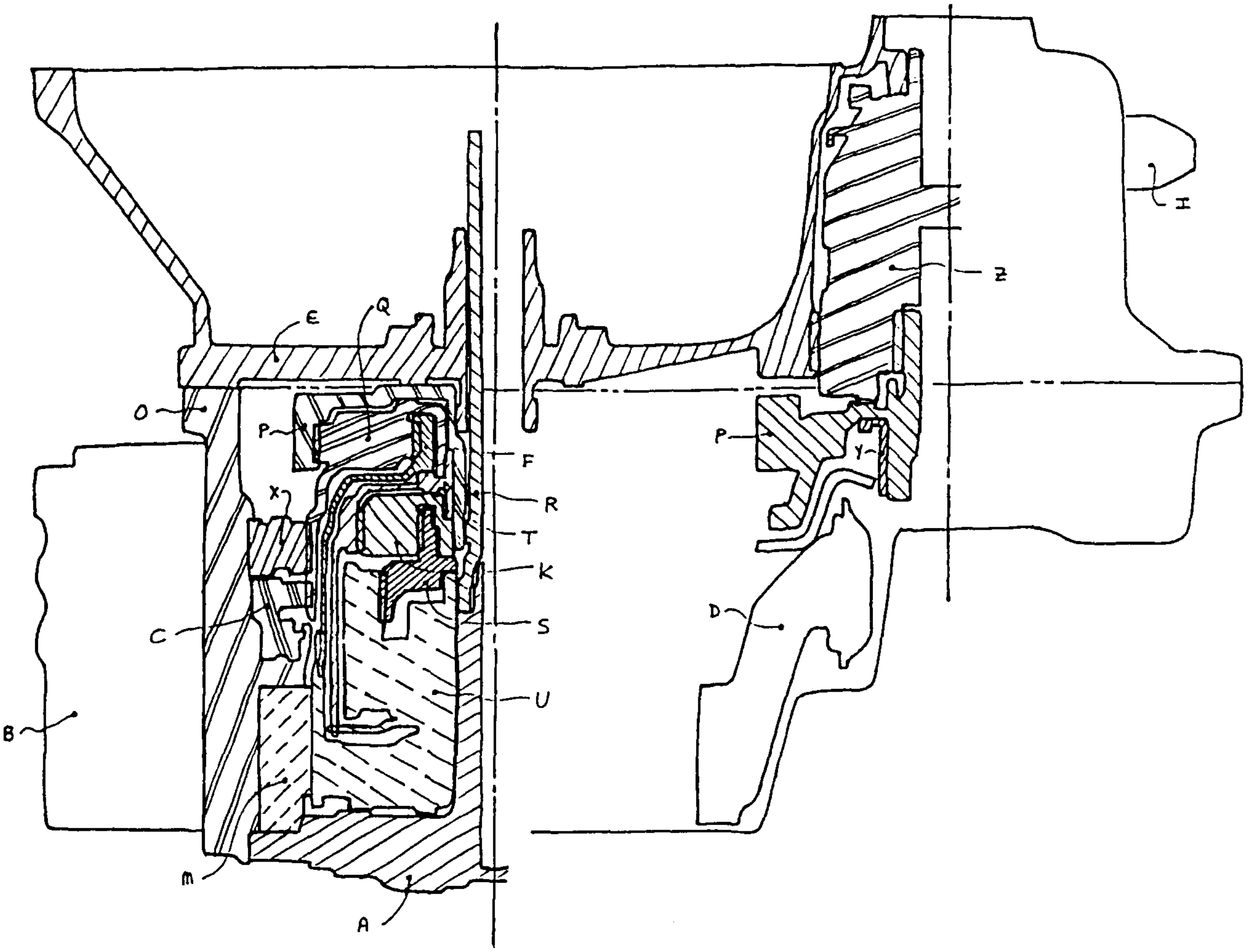
(D) BELL HOUSING
(B) INPUT SHAFT - C

(E) TORQUE CONUERTO

(M) End Bolt & Seal
(F) FRONT COUER

(M)ove component (C)omplete

Figure 13


E
Q
O
P
X
C
B
m
A
F
R
T
K
S
U
Z
P
Y
D

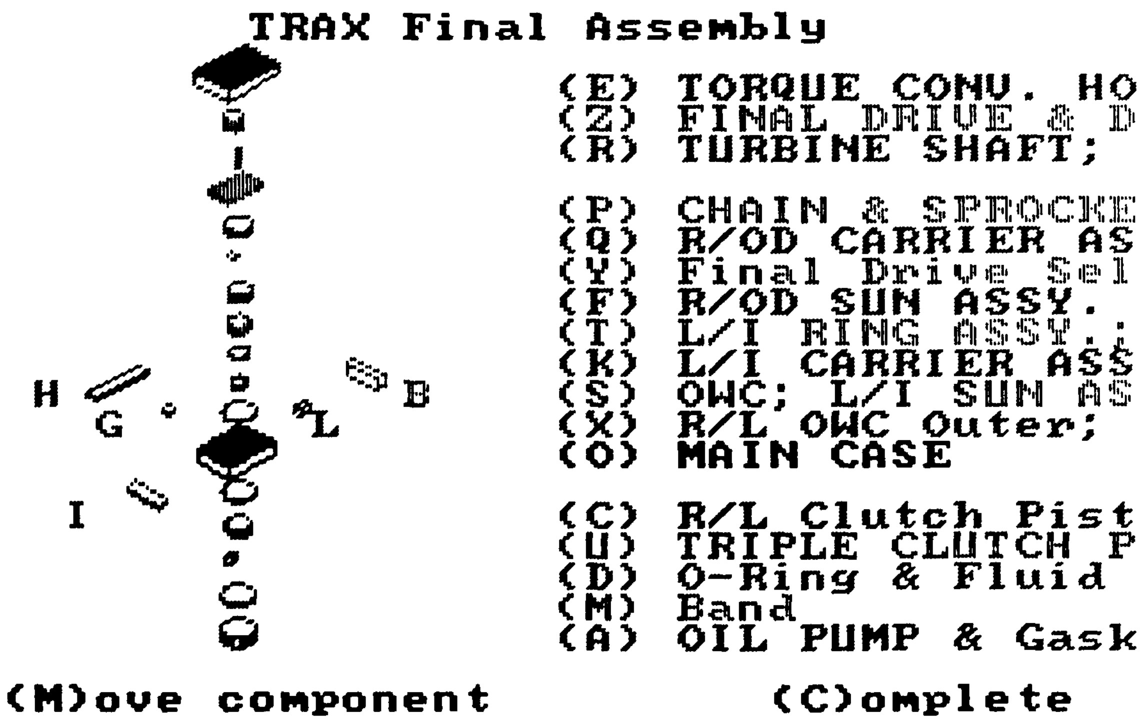
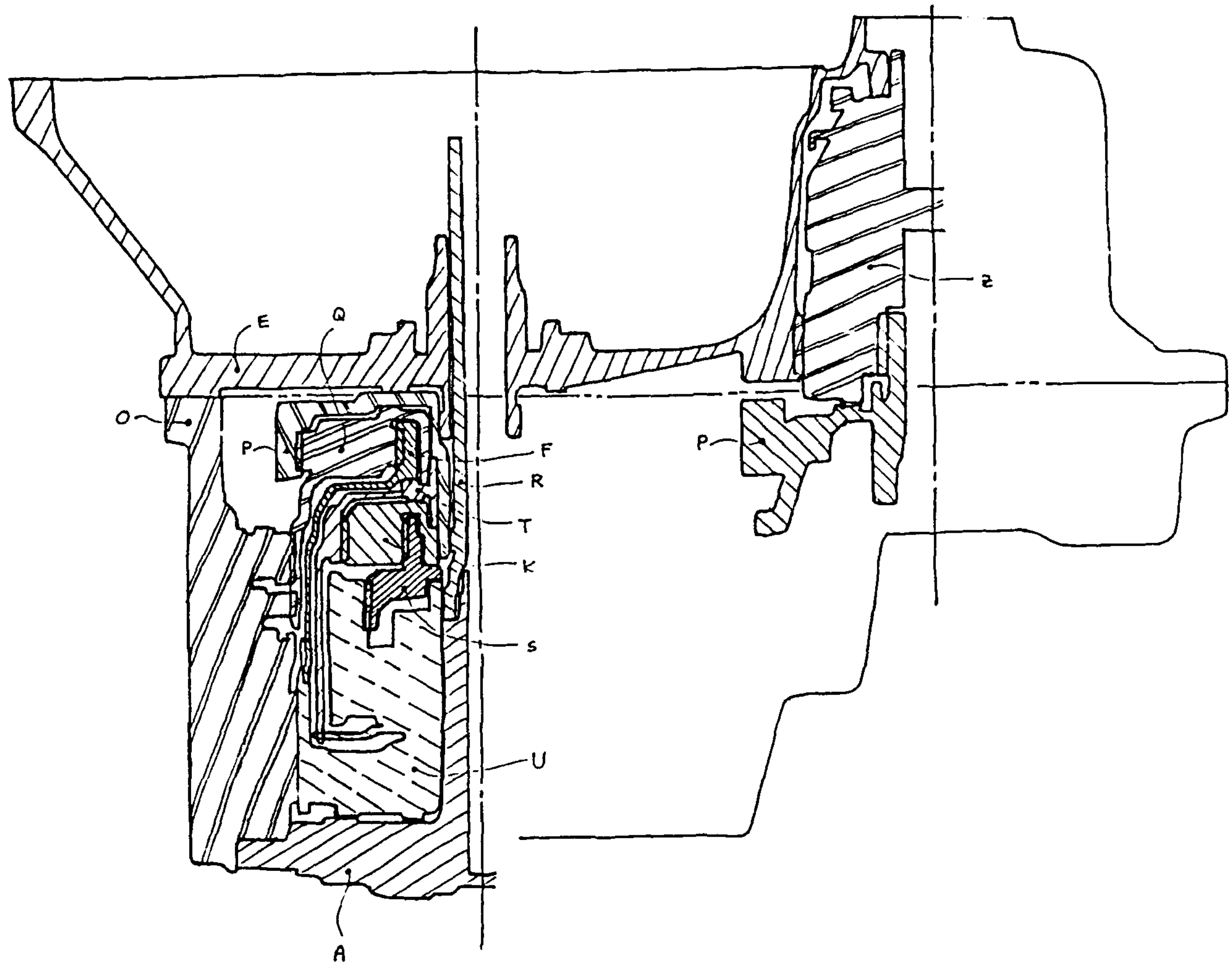

TRAX Final Assembly
(E) TORQUE CONV. HO
(Z) FINAL DRIVE & D
(R) TURBINE SHAFT;
(P) CHAIN & SPROCKE
(Q) R/OD CARRIER AS
(Y) Final Drive Sel
(F) R/OD SUN ASSY.
(T) L/I RING ASSY..
(K) L/I CARRIER AS$
(S) OWC; L/I SUN AS
(X) R/L OWC Outer;
(O) MAIN CASE
(C) R/L Clutch Pist
(U) TRIPLE CLUTCH P
(D) O-Ring & Fluid
(M) Band
(A) OIL PUMP & Gask
H
G
L
B
I
(M)ove component
(C)omplete

Figure 14


E
Q
O
P
F
R
T
K
S
U
A
P
Z

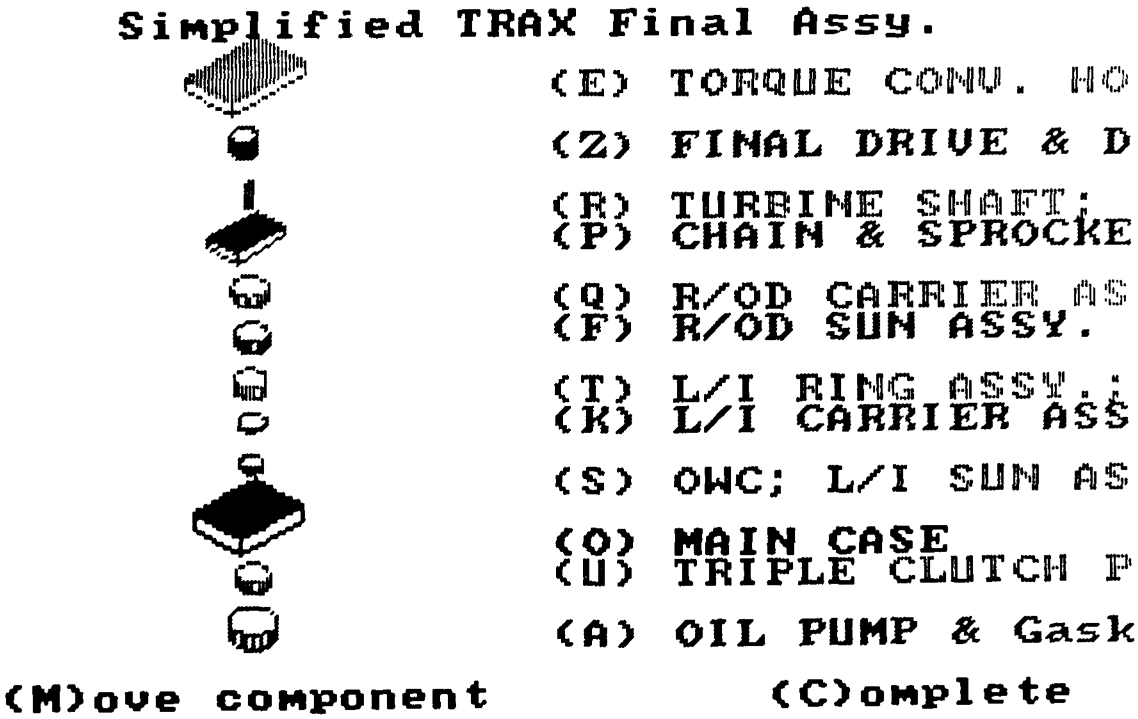
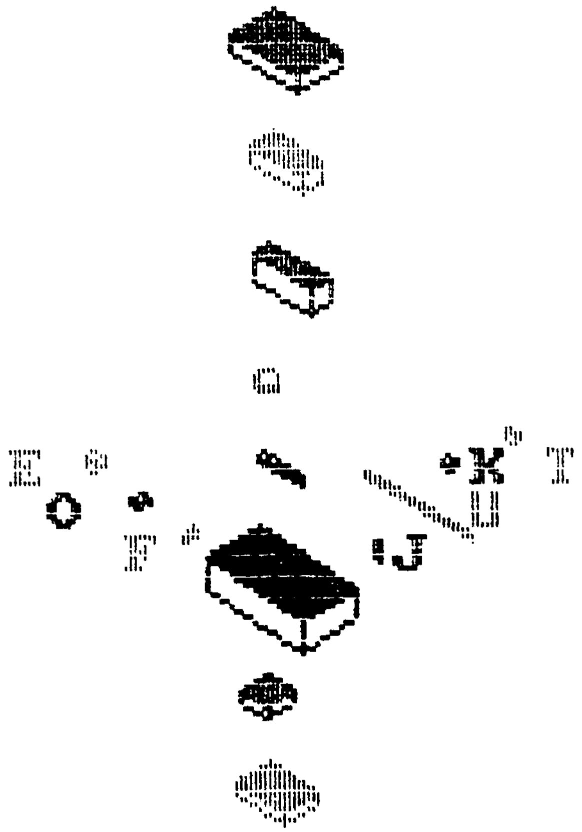

Simplified TRAX Final Assy.
(E) TORQUE CONV. HO
(Z) FINAL DRIVE & D
(R) TURBINE SHAFT;
(P) CHAIN & SPROCKE
(Q) R/OD CARRIER AS
(F) R/OD SUN ASSY.
(T) L/I RING ASSY.;
(K) L/I CARRIER ASS
(S) OWC; L/I SUN AS
(O) MAIN CASE
(U) TRIPLE CLUTCH P
(A) OIL PUMP & Gask
(M)ove component
(C)omplete

Figure 15

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:07:48



(A) Cover

(R) Heater Core; He

(N) Evap. Core Sub

(H) Motor & Fan; I

(S) Temperature Val

(L) Evaporator Case

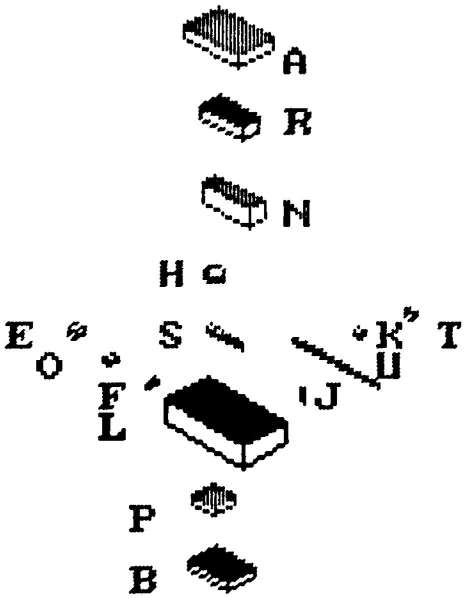
(P) AI Case; AI Val

(B) MVH Sub-Assy.

Sequence of assembly :

B P L S F O E U K T H N R A J

Figure 16



Base: L

B: P B

L: F O E

P: J U K T

T: S H N R A

Specify general order
(separate by commas):

Figure 17

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:16:14

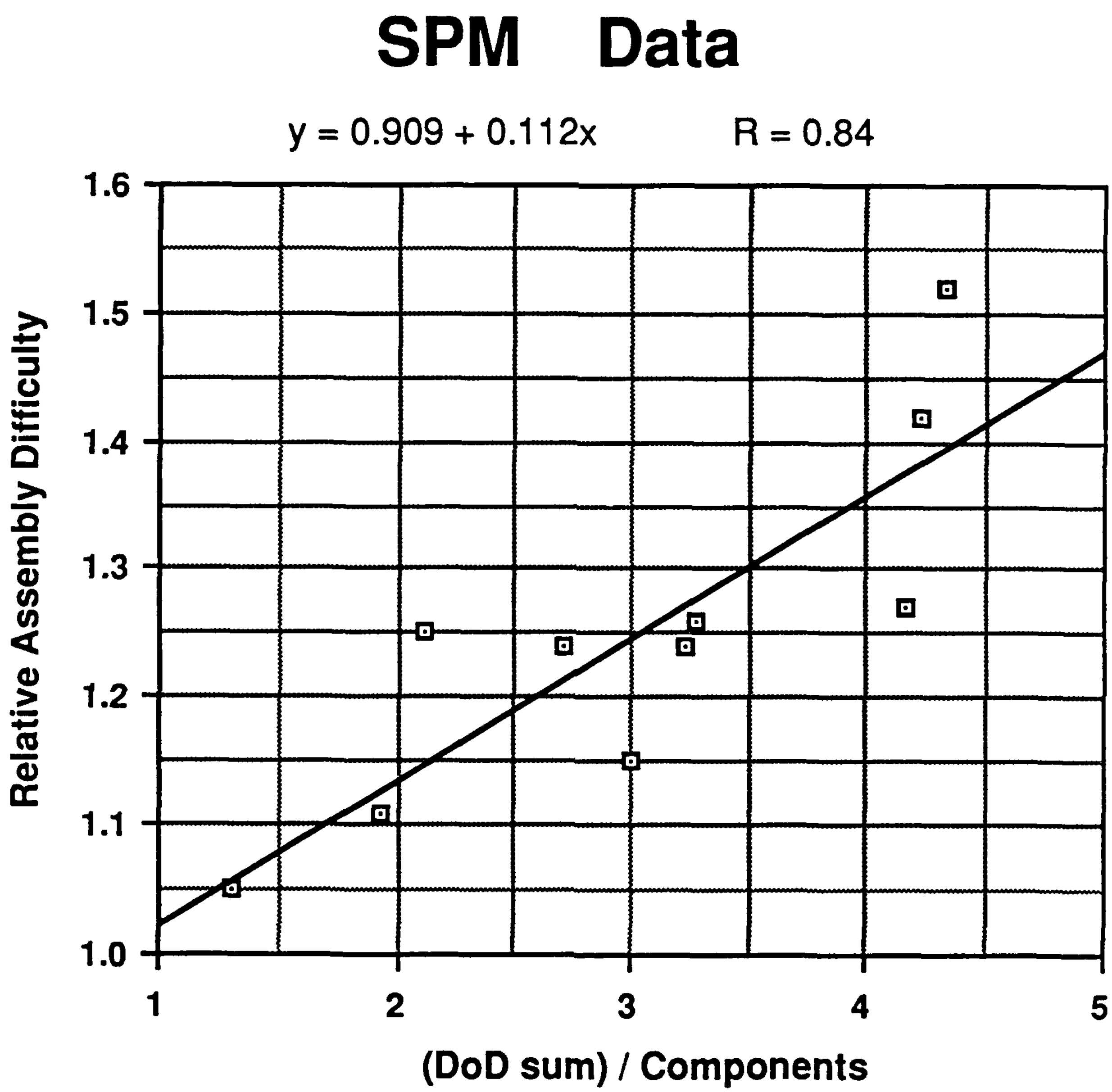
Envelope Size : X 15.6 Y 23.0 Z 11.5

Relative Assembly Difficulty 1.148

ask	Type	Motions Required	Load	Dgree of Diffclty	Task Actns	Task Description
1	P	Z	4.00	2	1	Attach MVH Sub-Assy. (B) to pallet.
2	P	Z	4.00	2	1	Position and attach fixture to pallet.
3	I	Z	4.00	2	4	Install AI Case; AI Valve; Vacuum Element; Link (P).
4	E	Z	5.00	1	1	Snap fit Evaporator Case (L) into assembly.
5	A	Z	1.00	3	1	Assemble Temperature Valve (S).
6	T	Y	2.00	2	1	Position Temp. Valve Actuator (F) and tighten fasteners.
7	T	Y	1.00	1	1	Position Solenoid #1 (O) and tighten fasteners.
8	E	Y	1.00	1	1	Snap fit Vacuum Element #2 (E) into assembly.
9	E	X	1.00	4	2	Snap fit Harness (U) into assembly.
10	M	X	0.00	2	1	Test assembled components.
11	T	X	1.00	1	1	Position Resistor Assy. (K) and tighten fasteners.
12	T	X	1.00	1	1	Position Solenoid Assy. (T) and tighten fasteners.
13	P	Z	4.00	2	1	Remove fixture.
14	P	Z	4.00	1	2	Place Motor & Fan; Isolator (H) into position.
15	P	Z	4.00	1	1	Place Evap. Core Sub-Assy. (N) into position.
16	P	Z	7.00	1	3	Place Heater Core; Heater Core Shroud; Clamp (R) into position.
17	A	Z	1.00	1	1	Align Cover (A).
18	I	Z	1.75	1	7	Insert bolts.
19	B	Z	14	2	7	Torque bolts.
20	L	X	1.00	4	1	Critical alignment of Pipe Seal (J) required.
21	V	X Y Z	0.00	2	1	Perform final test.
22	P	X Y Z	38	1	1	Pack / Unload assembly.

Task description data set name : ACMFA

Figure 18



SPM Data
y = 0.909 + 0.112x
R = 0.84
1.6
1.5
1.4
1.3
1.2
1.1
1.0
Relative Assembly Difficulty
1
2
3
4
5
(DoD sum) / Components

Figure 19

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:30:46

Envelope Size : X 15.6 Y 23.0 Z 11.5

Relative Assembly Difficulty 1.148

Task	Type	Motions Required	Load	Dgree of Diffclty	Task Actns	Task Description
1	P	Z	4.00	2	1	Attach MVH Sub-Assy. (B) to pallet.
2	I	Z	4.00	2	4	Install AI Case; AI Valve; Vacuum Element; Link (P).
3	E	Z	5.00	1	1	Snap fit Evaporator Case (L) into assembly.
4	A	Z	1.00	3	1	Assemble Temperature Valve (S).
5	T	Y	2.00	2	1	Position Temp. Valve Actuator (F) and tighten fasteners.
6	T	Y	1.00	1	1	Position Solenoid #1 (O) and tighten fasteners.
7	E	Y	1.00	1	1	Snap fit Vacuum Element #2 (E) into assembly.
8	E	X	1.00	4	2	Snap fit Harness (U) into assembly.
9	M	X	0.00	2	1	Test assembled components.
10	T	X	1.00	1	1	Position Resistor Assy. (K) and tighten fasteners.
11	T	X	1.00	1	1	Position Solenoid Assy. (T) and tighten fasteners.
12	P	Z	4.00	1	2	Place Motor & Fan; Isolator (H) into position.
13	P	Z	4.00	1	1	Place Evap. Core Sub-Assy. (N) into position.
14	P	Z	7.00	1	3	Place Heater Core; Heater Core Shroud; Clamp (R) into position.
15	A	Z	1.00	1	1	Align Cover (A).
16	I	Z	1.75	1	7	Insert bolts.
17	B	Z	14	2	7	Torque bolts.
18	L	X	1.00	4	1	Critical alignment of Pipe Seal (J) required.
19	V	X Y Z	0.00	2	1	Perform final test.
20	P	X Y Z	38	1	1	Pack / Unload assembly.

Task description data set name : ACMFA

Figure 20

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:31:48

1.50 LOAD Safety Factor 1.20 REACH Safety Factor

Task specification data set name : ACMFA

	MANUAL	PROGRAMMABLE	FIXED AUTOMATION
Cost Weight :			
LOAD	32.2%	38.4%	23.9%
REACH	27.5%	23.2%	9.2%
DoF	40.3%	38.4%	66.9%
Time Factor	1.000	0.900	0.500

Task	Resource Cost	Tooling Cost	Task Time	Tool Number	Resource Cost	Tooling Cost	Task Time	Tool Number	Resource Cost	Tooling Cost	Task Time	Tool Number
1	200	2600	11.5s	1	50500	11000	10.5s	1	0	134000	6.0s	1
2	200	2600	19.5s	2	50500	11000	17.5s	2	0	134000	10.0s	2
3	200	1800	14.0s	3	37500	8000	12.5s	3	0	102500	7.0s	3
4	200	3100	13.5s	4	47500	11500	12.5s	4				
5	200	3000	19.0s	5	51500	11000	17.0s	5	0	209500	9.5s	4
6	200	2100	25.0s	6	33000	7500	22.5s	6	0	140500	12.5s	5
	200	2100	25.0s	7	33000	7500	22.5s	7	0	140500	12.5s	6
8	200	3800	27.0s	8								
9	200	2700	14.0s	9					0	128500	7.0s	7
10	200	1900	18.0s	10	29500	7000	16.0s	8	0	95000	9.0s	8
11	200	1900	18.0s	10	29500	7000	16.0s	8	0	95000	9.0s	9
12	200	1800	18.0s	11	35500	7500	16.5s	9	0	95000	9.0s	10
13	200	1800	14.0s	11	35500	7500	12.5s	9	0	95000	7.0s	11
14	200	1800	21.0s	11	40500	8500	19.0s	10	0	117500	10.5s	12
15	200	1800	14.0s	12	27500	6500	12.5s	11	0	72500	7.0s	13
16	200	1800	29.0s	13	30000	7000	26.5s	12	0	77500	7.6s	14
17	200	39000	19.5s	14	67500	13000	17.5s	13	0	179000	10.0s	15
18	200	3800	21.0s	15								
19									0	278500	31.5s	16
20	200	2800	32.5s	16	88500	14500	29.0s	14				

Figure 21

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:34:16

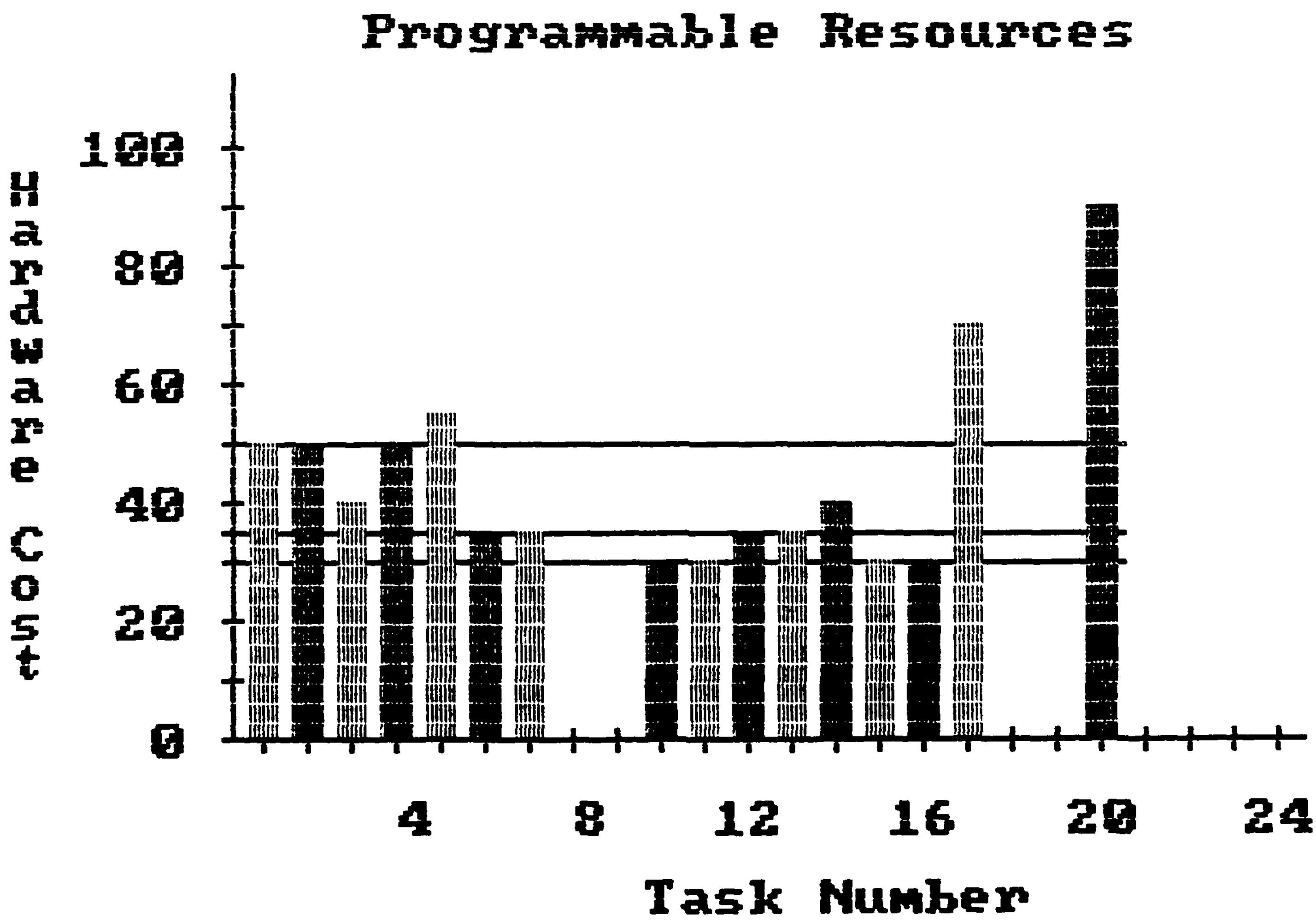


Figure 22

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PREPROCESSOR 04-03-1990 10:33:00

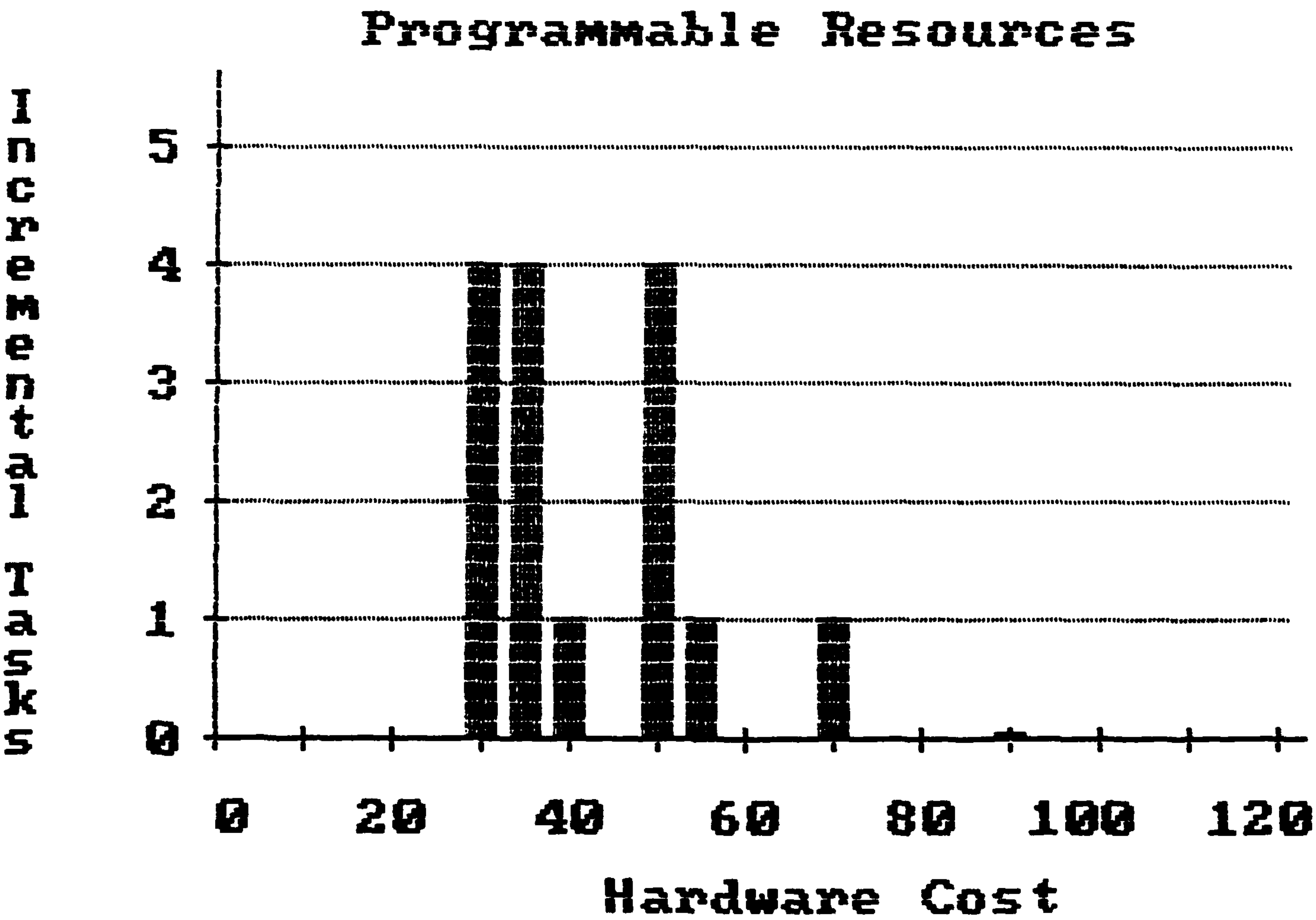


Figure 23

AIR CONDITIONING MODULE Final Assy.

ASSEMBLY SYSTEM DESIGN PROGRAM DATA 03-30-1990 14:27:37

Results of ASDP pre-processor

Resource Data Set name : ACMRES
Task Data Set name : ACMTSK

RESOURCE	MNL	FXD	P30	P35	P50
Hardware cost ($)	200	0	30000	35000	50000
(Total cost) / (Hardware cost)	1.20	1.00	1.75	1.88	2.25
% up-time expected	88.49	94.22	94.22	94.22	94.22
Operating/maintenance rate ($/hr)	0.50	1.16	0.60	0.70	1.00
Tool change time (seconds)	2.0	0.0	4.0	4.3	5.0
Maximum stations per worker	0.83	6.13	5.89	5.65	5.00

TASK	Description	MNL	FXD	P30	P35	P50	TASK
1	Attach MVH Sub-Assy. (B) to pallet.	12.0s : 101 / 2600	6.0s : 201 / 134000			10.5s : 501 / 11000	1
2	Install AI Case; AI Valve; Vacuum Element; Link (P).	20.0s : 102 / 2600	10.0s : 202 / 134000			18.0s : 502 / 11000	2
3	Snap fit Evaporator Case (L) into assembly.	14.5s : 103 / 1800	7.5s : 203 / 102500			13.0s : 503 / 8000	3
4	Assemble Temperature Valve (S).	14.0s : 104 / 3100				12.5s : 504 / 11500	4
5	Position Temp. Valve Actuator (F) and tighten fasteners.	19.5s : 105 / 3000	10.0s : 204 / 203500				5
6	Position Solenoid #1 (O) and tighten fasteners.	26.0s : 106 / 2100	13.0s : 205 / 140500		23.5s : 406 / 7500	23.5s : 506 / 7500	6
7	Snap fit Vacuum Element #2 (E) into assembly.	26.0s : 107 / 2100	13.0s : 206 / 140500		23.5s : 407 / 7500	23.5s : 507 / 7500	7
8	Snap fit Harness (U) into assembly.	28.0s : 108 / 3800					8
9	Test assembled components.	14.5s : 109 / 2700	7.5s : 207 / 128500				9
10	Position Resistor Assy. (K) and tighten fasteners.	18.5s : 110 / 1900	9.5s : 208 / 95000	16.5s : 308 / 7000	16.5s : 408 / 7000	16.5s : 508 / 7000	10
11	Position Solenoid Assy. (T) and tighten fasteners.	18.5s : 110 / 1900	9.5s : 209 / 95000	16.5s : 308 / 7000	16.5s : 408 / 7000	16.5s : 508 / 7000	11
12	Place Motor & Fan; Isolator (H) into position.	18.5s : 111 / 1800	9.5s : 210 / 95000		17.0s : 409 / 7500	17.0s : 509 / 7500	12
13	Place Evap. Core Sub-Assy. (N) into position.	14.5s : 111 / 1800	7.5s : 211 / 95000		13.0s : 409 / 7500	13.0s : 509 / 7500	13
14	Place Heater Core; Heater Core Shroud; Clamp (R) into position.	22.0s : 111 / 1800	11.0s : 212 / 117500			19.5s : 510 / 8500	14
15	Align Cover (A).	14.5s : 112 / 1800	7.5s : 213 / 72500	13.0s : 311 / 6500	13.0s : 411 / 6500	13.0s : 511 / 6500	15
16	Insert bolts.	30.5s : 113 / 1800	8.1s : 214 / 77500	27.5s : 312 / 7000	27.5s : 412 / 7000	27.5s : 512 / 7000	16
17	Torque bolts.	20.0s : 114 / 39000	10.0s : 215 / 179000				17
18	Critical alignment of Pipe Seal (J) required.	21.5s : 115 / 3800					18
19	Perform final test.		32.5s : 216 / 278500				19
20	Pack / Unload assembly.	33.5s : 116 / 2800					20

LEGEND : Operation time : 'Tool' Number / Hardware Cost ($)

Figure 24